Write-up for the Diffractometer D1 at Risø

Bundgaard, Jørgen; Krebs Larsen, F.; Lebech, Bente; Nielsen, M. H.; Skaarup, Per

Publication date:
1982

Document Version
Publisher's PDF, also known as Version of record

RISØ-M-2344

WRITE-UP FOR THE DIFFRACTOMETER D1 AT RISØ

Ed. by J. Bundgaard, F. Krebs Larsen, B. Lebech,
M.H. Nielsen and P. Skaarup

Abstract. Manual for the crystallographic program system used to control the 4-circle neutron diffractometer D1/TASII at DR3, Risø. The mechanical part of the diffractometer consists of a monochromator part which allows an easy change of incident neutron wavelength and a four-circle HUBER goniostate consisting of an Euler cradle (HUBER 512) and two horizontal goniometers (HUBER 440 and HUBER 430). The goniostate is computer controlled by a PDP-11/34 interfaced via CAMAC modules. The PDD-11/34 computer has a 128 k byte memory, two hard magnetic disc stations, a fast DEC-writer terminal and a screen terminal. The diffractometer can be operated remotely via modem and telephone line connections from remote stations such as the University of Århus and ILL, Grenoble. Minor parts of the software used to control the diffractometer were developed at Risø while the major parts were a generous gift to Risø from College 5, the diffraction group, at the Institute Laue-Langevin, Grenoble, France.

INIS descriptors: COMPUTER CODES; CRYSTALLOGRAPHY; MANUALS; NEUTRON DIFFRACTOMETERS; PDP COMPUTERS

UDC 548.7 : 681.3.08

May 1982
Risø National Laboratory, DK 4000 Roskilde, Denmark

The D1 4-circle diffractometer (TAS II) installed at DR3 at RISOE was financed by the Danish Natural Science Research Council with contributions from Chemical Institute, University of Aarhus and Risoe National Laboratory. Large parts of the software was a gift from Institute Laue-Langevin, Grenoble.

The diffractometer is administered by the Danish National Comittee for Crystallography.
Potential users can adress either

Bente Lebech, Fysikafdelingen
Forsoegsanlaeg RISOE
4000 Roskilde Tlf. 02371212
or
Finn Krebs Larsen, Kemisk Institut
Aarhus Universitet, Langelandsgade 140
8000 Aarhus C Tlf. 06124633

Pour qu'une chose soit interessante
il suffit qu'on la regarde longtemps.
Gustave Flaubert

Prepared originally for ILL D8 and D9 by
J.Allibon and M.S.Lehmann

Modified for RISOE D1 (TASII) May 1981 by
J.Bundgaard, F.Krebs Larsen, and P.Skaarup

Revised version May 1982 by
J.Bundgaard, F.Krebs Larsen, B.Lebech,
M.H.Nielsen and P.Skaarup

ISBN 87-550-0839-9
ISSN 0418-6435

Risø Repro 1982

CONTENT

ACKNOWLEDGEMENTS

The ILL-authors of this report writes:
Our thanks to A.Barthelemy, A.Filhol, P.G.Rice, C.Turfat, S.A.Mason, J.Pannetier and M.Thomas for their efforts in building and improvements of the system.

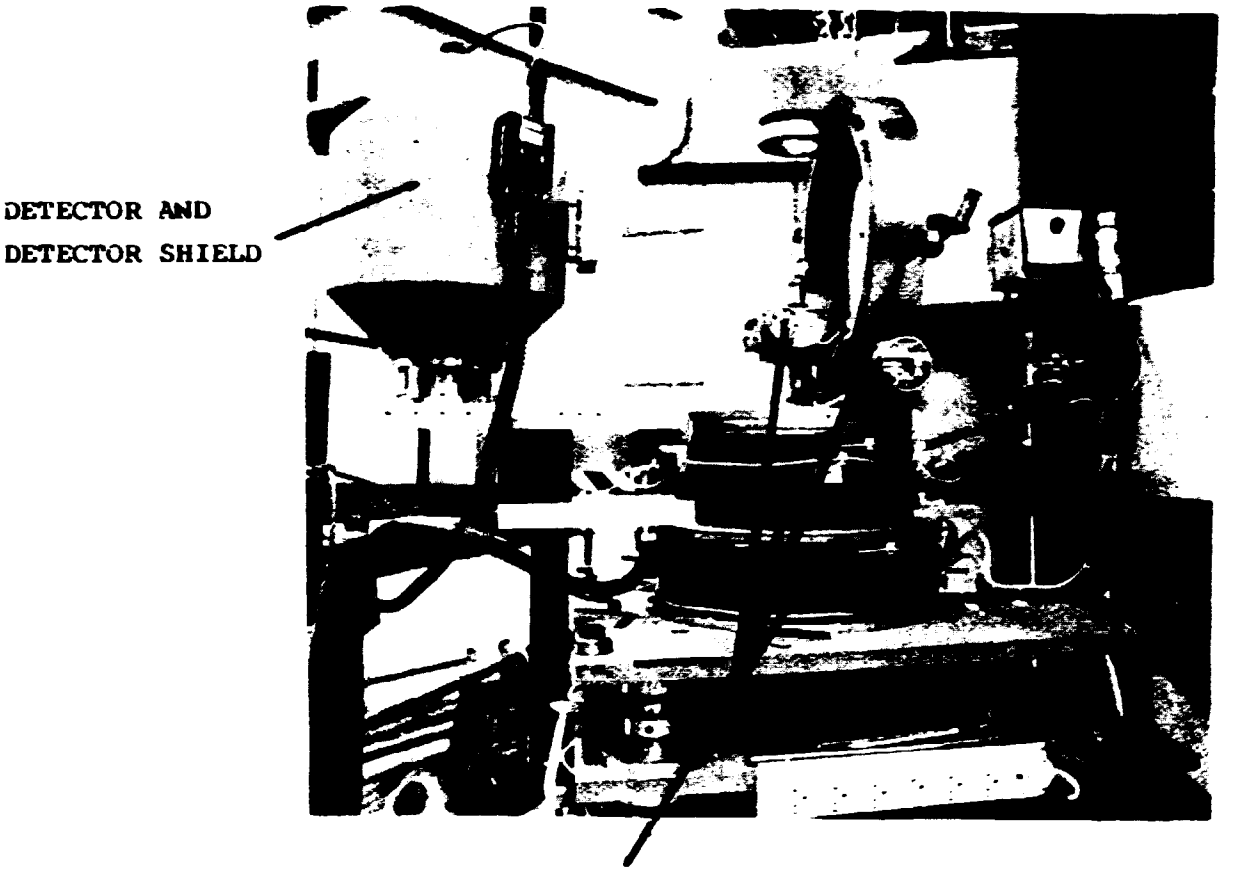
Our special thanks are extended towards S.A.Mason for his constructive criticism and his help in our minor and sometimes major spelling crises during the building of this manual.

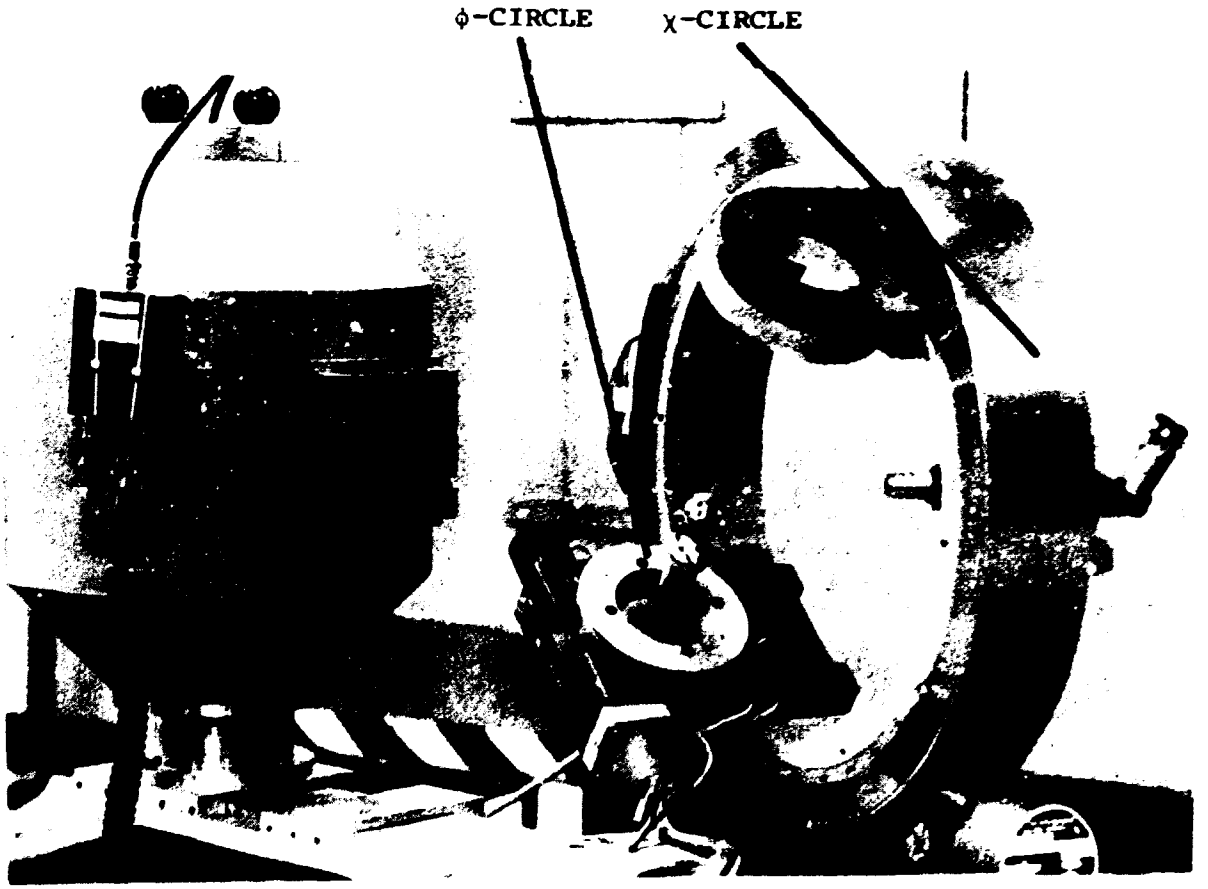
The danish authors of this report writes:
The staff at the Chemical Institute, University of Aarhus and Risoe Natinal Laboratory are grateful to the staff of college 5 at the Institute Laue Langevin, France, for allowing the use of large parts of the diffractometer software and the crystallografic data analysis programs developed at ILL. Special thanks are due to Dr.M.Lehmann for his cooperation and the help in getting the diffractometer into operation.

DISCLAIMER

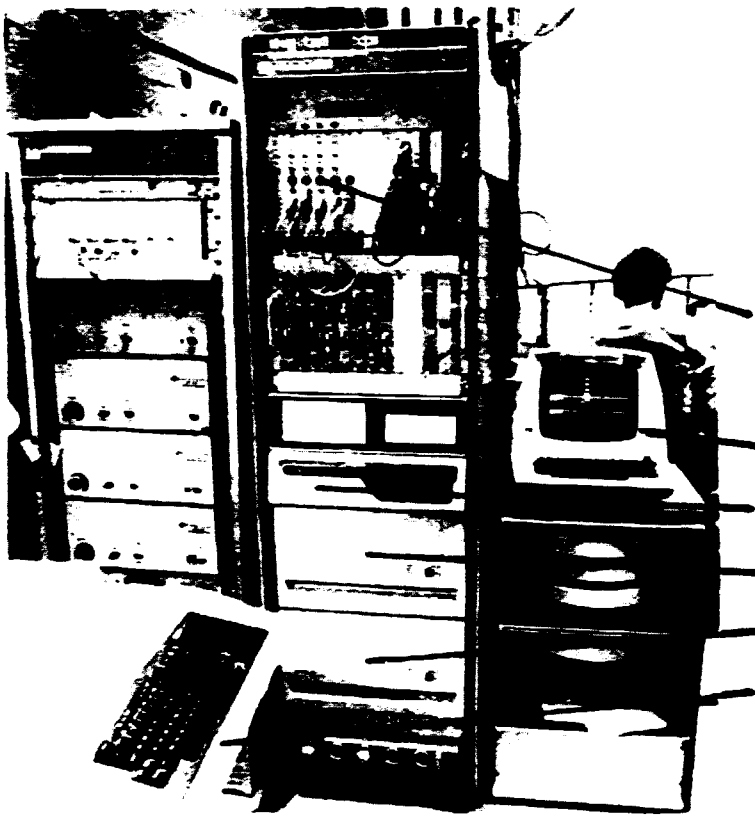
Although much has been done to ensure that this note is correct and up-to-date, the user is asked to take all the necessary care when using the system. Indeed we would be very happy to get suggestions for improvements that do not unduely increase the extent of the manual.

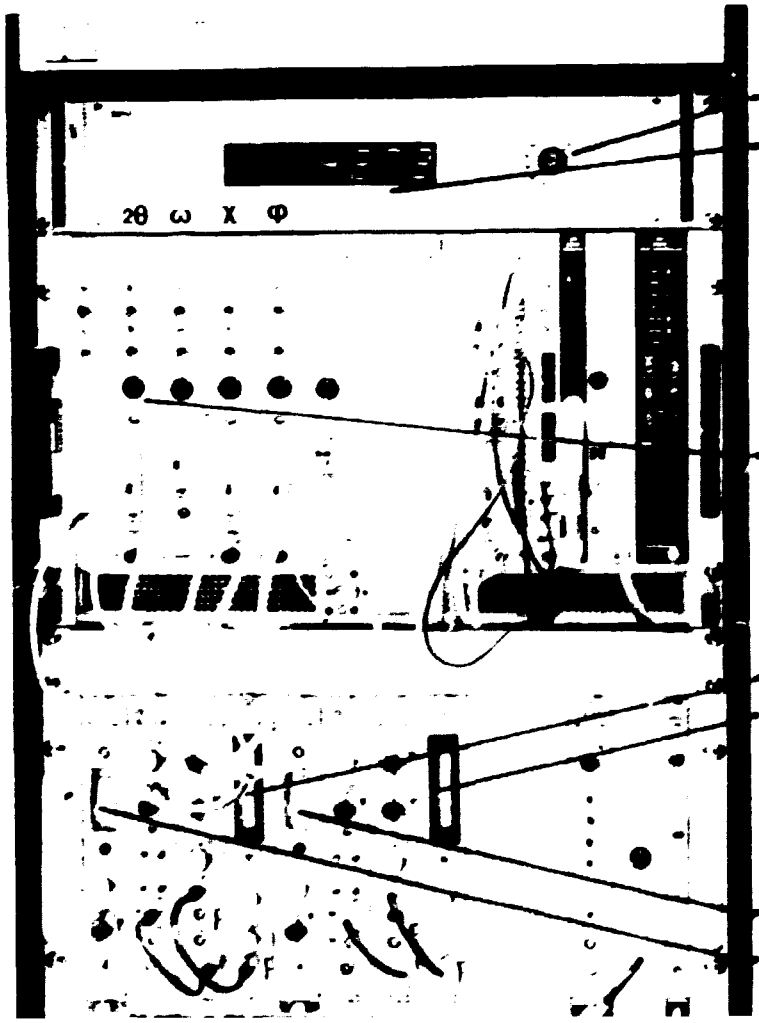
MECHANICAL PARTS OF THE 4-CIRCLE DIFFRACTOMETER AT DR 3



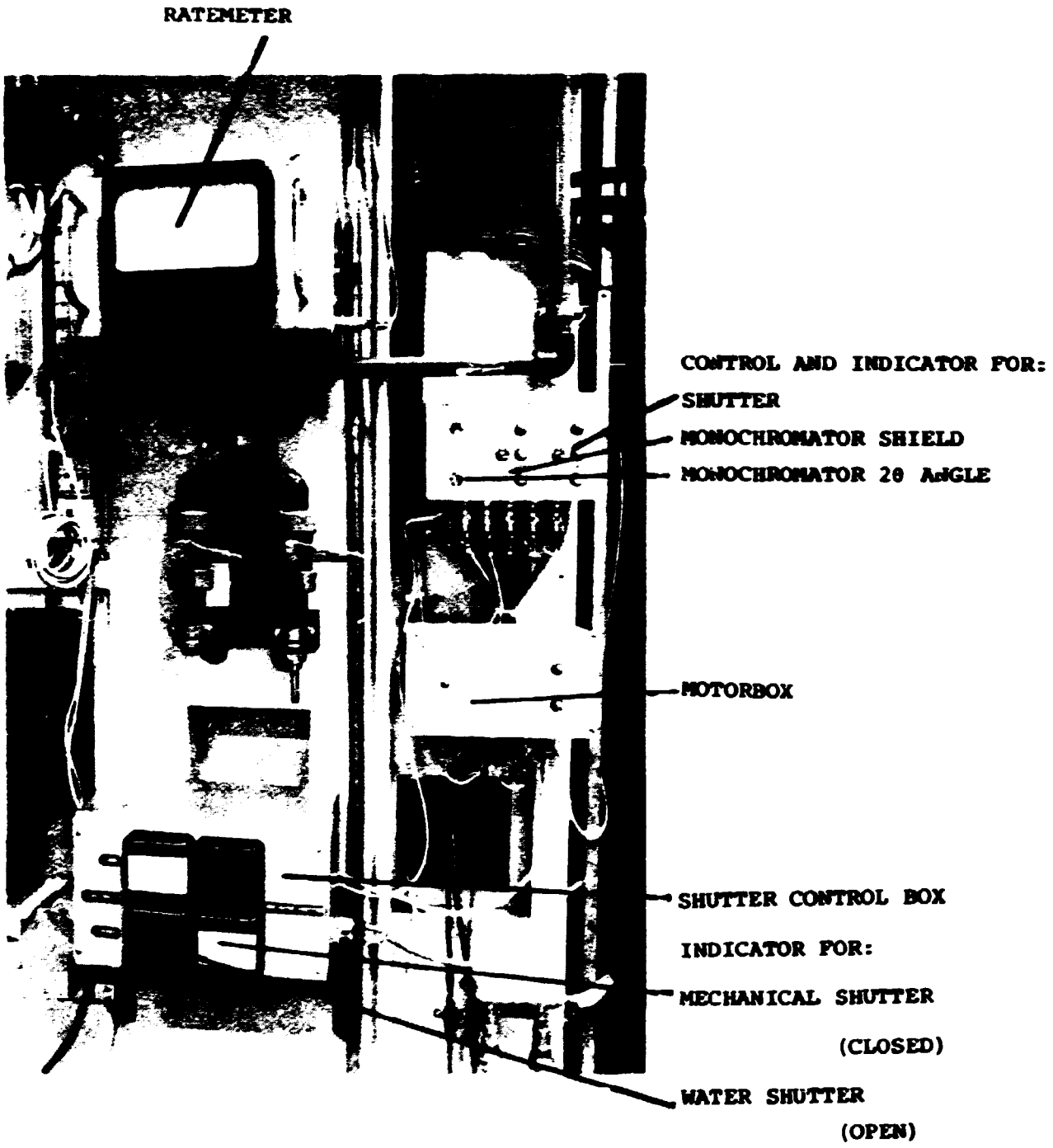


ELECTRONICS FOR THE 4-CIRCLE NEUTRON DIFFRACTOMETER





4-CIRCLE DIFFRACTOMETER AT DR3



IMPORTANT INDICATOR- AND CONTROL-BOXES
AT THE DIFFRACTOMETER PLATFORM

CHAPTER 1

INTRODUCTION

These pages serve as a short guide for the use of the program systems on D1. Appended are notes on the instruments and environment control units. If everything fails then more elaborate manuals are available at the instrument.
Some of these are:

PDP-11 manuals

A.Barthelemy and A.Filhol:
Systeme LSD. Ill Report 78BA51T

A.Barthelemy, P.G.Rice and C.Turfat:
Systeme D8 -PDP 11. Ill Report 78BA72T

A. Filhol and M.Thomas:
Manuals for some routine programs for X-ray and neutron 4-circle diffractometers and for crystallography. Ill Report 76F291T

M.S.Lehmann and S.Wilson:
College 5 data reduction system. Ill Report 74L111T

The present note aims at giving the information necessary for a measurement. It is hoped that it will inspire the user, as time goes by, to familiarize himself with all the advanced features that are available , which are described in the above manuals and notes.

CHAPTER 2

THE COMPUTER CONFIGURATION AND SYSTEM

2.1 FIRST IMPRESSIONS

The computer is a PDP-11/34. The operating system is RSX-11-M and the memory size is 128K. This allows several programs to turn simultaneously , and each terminal can be in contact with several programs at any time.
There are one fast decwriter, a screen terminal and two modem lines. The decwriter is the system terminal and is used to run LSD, i.e. the measurement.

To return to monitor level press the the RETURN key and the system will answer
>
Then give the desired command. Interactions with the system are normally of the type
>AAA BIMBAM
The arrow is always given by the monitor. AAA is a command (e.g. RUN, TEC, LSD) and BIMBAM is a series of parameters including file or program names (e.g. FOR004.DAT, COLL5N).

To stop (brutally) a program XXX write
>ABO XXX
The arrow is given by the monitor, ABO XXX by the user.
(If any file, e.g. TOTO.DAT was beeing written to or from by the program it will be left locked. Unlock it by the command
>OPN TOTO.DAT).

For program LSD a smooth stopping can be done using PCP (see A-15), and for COLL5N, KCO is used (for details see 8.8).

To see the active programs running on the system type
>ACT /ALL
and the monitor will answer, for example
. LDR.
...MCR
...SYS
F11ACP
...LSD
XXX

In this case LSD and XXX are active. They can be stopped as described above and below.

. LDR. MCR SYS and F11ACP are system programs which are always present. Never try to abort them.

Examples of other interactions are
>TIM
which gives the date and time.
>LSD
which runs the scan and measurement program.
Never run LSD from more than one terminal at the same time.
>RUN XXX
starts program XXX. In many cases the program will ask questions.

2.2 USEFUL CNTRL CHARACTERS

Press the key CTRL and C : Gives access to the monitor whilst a program is running and giving output.
Press the key CTRL and U : Erase line you are in the process of typing.
Press the key CTRL and O : Silences terminal but output still continues:-- Next CTRL/O restarts the output on terminal.
Press the key CTRL and Z : Return to monitor level from system programs such as PIP.
Press the key CTRL and S : Pauses listing of output. (and holds program, unlike CRTL/O)
Press the key CTRL and Q : Restart listing after CTRL/S.
Press the key CTRL and R : Re-types the current input line corrected after rubouts etc.

2.3 DISKS

There are two disks.

DL0 contains system programs, LSD and other common programs as well as data files for these.
DL1 contains measured reflections and some backup files.

Most user operations are done on DL0, which is the default disk. It need therefore not be mentioned in the command. If we work on DL1: we must indicate the source.

Example:
>TEC DL1:SOLEIL.DAT
We want to correct something (using the system program TEC) in the file SOLEIL.DAT which is found on DL1. If the file was on DL0 the command would be
>TEC SOLEIL.DAT

2.3.1 ASSIGNING OF DISKS

For convenience, three dummy devices are used by the programs described later in the manual. These are assigned to various disks depending on what you wish to do.

DI1: is the device to which you write data.

TR1: is the device from which data is transferred to the B7800

DR1: the device from which you do non-automatic data reduction.
Note - automatic data reduction is done from DI1:

The command ASN is used to change or examine assignments.

To look at current assignments, before changing them, do

ASN /GBL

and you will see the current status of DI1: TR1: DR1:

To change, for example, TR1: to DL1: do

ASN DL1:=TR1:/GBL

Normal running will mean writing data on DL1: (DL1:=DI1:)
Data transfer from DL1: (DL1:=TR1:)

2.3.2 WORKING AREAS [uic 's].

There are various working areas; The one available for the user is numbered [210,210]. Do not use any other area. To check that this area is attached to the terminal do
>LOG
and the system should answer
[210,210]
If not, command
>LOG 210,210

Area [200,200] is for programs.

2.4 LOGGING ON TO THE SYSTEM

The decwriter should stay logged onto the system. There should never be any need to use the system commands

HEL (to log-on)

and

BYE (to log-off)

on the decwriter.

The modem lines must be logged on

HEL D1/D1

This logs you into the number 7,7. This is an empty directory but, by coming into the system here, you are given all the necessary privileges to run the system.
Then go to your normal working area [210,210] by typing

LOG 210,210

and

MOU DL1:/OVR, if DL1 is to be used.

CHAPTER 3

COMMANDS FOR FILE HANDLING

3.1 FILES

The complete name for a file under RSX would be

DEV:[uic]FILENAME.EXT;VERSION No

An example might therefore be

DL1:[200,200]ADVENT.FTN;15

DL1: States on which device the file is found. Default DL0:

[200,200] is the user area where it is found. Default [210,210]

ADVENT is a label, -any possible name up to 6 characters.

FTN Says what type of file it is. examples are
FTN - fortran source file,
OBJ - machine compiled file,
TSK - machine linked task file (runnable program),
DAT - input/output-data file,
CMD - command string file,
BAK - backup file,
etc,etc.

The final number is the version number.
Each modification to the file produces a new file with the same name but hisner version number. Note that the old version still exists on the system. The computer will, by default, look at the most recent version.

Therefore if one types ADVENT.FTN
the computer assumes you to be talking about the most recent version of ADVENT.FTN on DL0: [210,210].

A further example:
At some given time a list of files may be
RAFIN.DAT;237
RAFIN.DAT;240
RAFIN.DAT;241
When referred to as RAFIN.DAT it is RAFIN.DAT;241 that is assumed. Note, the version No. is in octal.

One can still access the older files by using their full name.
Example:
TEC RAFIN.DAT;237
Correct on RAFIN.DAT;237 rather than RAFIN.DAT;241

If you want to check the list of files, use the command
>DIR *.DAT
This will give all files (latest version) with extension DAT on the area where you are working, normally 210,210 on disk DL0:.

DIR PLONK.DAT;*
Gives you a list of all versions of the file PLONK.DAT

If you only want to keep the latest of a series, the command PUR PLONK.DAT will delete all but the latest version. Therefore the command
>PUR TOTO.DAT
will leave only the latest version of the file TOTO.DAT. The disk can fill up quite rapidly, so purge from time to time on the data files.

3.2 A GENERAL COMMAND TO THE MONITOR

A general form of a command under RSX is

>AAA DEV:[uic]NAME.EXT;VERSION NO.

Example :

>TYP DL1:[200,200]MAN.DOC

AAA is the command, TYPe in the example.
DEV, the device will normally be one of the disks, DL1: in the example. The default device is DL0:, i.e. if no DEV is specified, the system looks only at DL0:.
[uic] is the working area. It will normally be be either [200,200] or [210,210], Its default is the area in which you are working which should be [210,210].
NAME.EXT;VERSION No. is the file on which you wish to operate. If the VERSION No. is not given, the command automatically acts on the latest version.

Certain extensions are default for certain commands, examples:
FOR (Fortran compiler) command looks for a file with extension FTN.
RNO (Runoff programme) command looks for a file with extension RNO.

In its simplest form a command might be

>AAA NAME.EXT

Example :

>TYP FOR002.DAT
This says type the file FOR002.DAT

3.3 THE JOKER PARAMETER.

One very useful parameter is ' * '. This is the JOKER parameter. It can, under certain circumstances, replace all possible names. So the command

>DIR *.DAT
would give a list of all files with extension DAT on DL0:

>PUR FOR004.DAT;*
would purge all files but the latest version with name FOR004.DAT on DL0:

>PUR DL1:HANSEN.DAT;*
would purge all files but the latest version with name HANSEN.DAT on DL1:

NOTE

All RSX 11 commands consist of three letters. In most cases, any additional letters will give an error.

3.4 BACKING-UP FILES.

Important data files can be backed-up by writting them onto DL0:
If you do this, please rename them so that they are easy for us to identify.
They will normally be kept during your measurement, unless you specifically delete them, but the disks are usually cleaned up between measurements.

Copy your file onto DL0: with the command

```
>COP DL0:MYNAME.EXT=DL1:LISTOF.HKL
                              for example.
```

This copies the contents of the file LISTOF.HKL on DL1 into a file on DL0:, name it MYNAME.EXT

This back-up is mainly of interest if DL1: is changed during measurement

3.5 THE RUN COMMAND

The command

```
>RUN TASKXX
```

tells the monitor to run the linked program TASKXX.TSK

If this program (task) is part of the D1 software it will have been 'INSTALLED'. This means that details of the task have been written in the ' System Task Directory ' (STD) with the command INS.
Therefore, when you say RUN HKLGEN, the system looks in the STD for an entry for HKLGEN. Hopefully, it finds details of a task called HKLGEN which is contained on DL0: under uic [200,200], which it will then run for you.
If the entry is not found, it will look on the default device (DL0:) under the default directory (usually [210,210]) for a file HKLGEN.TSK which it will run for you.
If this is not found, it will give the message FILE NOT FOUND.

Tasks can be installed such that the RUN command is not needed. LSD is an example of this (see also p.D-4).
When LSD is typed into the monitor, the STD is examined for an entry under LSD.
Hopefully, the entry is found and the monitor is pointed at a file
DL0:[200,200]DECUSS.TSK, which is run.

All tasks present on DL0: (examples: LSD, COLL5N, TEC) are installed automatically when the system is bootstrapped.

3.6 THE ACTIVE TASK LIST.

The ACTIVE TASK LIST shows what is running on the system.
It can be examined by typing
>ACT
to see all jobs connected to this terminal, or
>ACT /ALL
to see all jobs running on the system.
(...LDR MCR F11ACP are system tasks which are always resident)

An installed task which is run from the ' RUN TASKNAME ' command, for example, RUN RAFIN, has an entry in the ATL of RAFIN.
It can be aborted, from any terminal, by typing ABO RAFIN.
Such tasks can only be running from one terminal at any time, subsequent requests will give the message TASK ACTIVE.

An installed task which is run from the three letters only option, for example FOR, has an entry in the ATL of ...FOR.
It can be aborted, from any terminal, by typing ABO ...FOR.
(the terminal from which it is running need only type ABO FOR)
Any number of terminals can call FOR, but note however that the ATL distinguishes between the different FOR's active at the same time by calling the first, ...FOR and the subsiquent requests, .FORTn where n is the terminal number.

A non-installed task, which can only be run as in ' RUN USERTASK ' will have an entry in the ATL as the terminal name from which it is run.
e.g. a game run on TT3: will be called TT3 and is aborted with ABO TT3.

Therfore, before trying to abort a task, always do an ACT /ALL to see what is running and what it is called by the system.

Example: An ACT /ALL is done and the system shows
...LDR
MCR...
...MCR
F11ACP
.LSDT2
T11V10
...AT.
TT1

An explanation of what is happening could be:
The first four are system tasks, ignore them.
LSD has been called somewhere on the system, and while it was active, a second LSD was called on TT2:, thus called .LSDT2 by the system to distinguish it from the original - which has subsequently been terminated.
T11V10 is running
also an unknown task from TT1 (probably an uninstalled game)
...AT. is explained in the next section.

3.7 RSX COMMAND FILES

Any file with an extension CMD is a command file.
It can be used to give a series of commands to the monitor as if from a terminal. It also has a possibility to ask the user questions and, depending on the reply, branch within itself etc.
Examples of command files used frequently on our system are

[1,2]STARTUP.CMD - the file called immediately after a bootstrap. It sets the time and sets up the system.

GAMES.CMD - asks you which game on the games disk you want to play and runs it for you.

A command file is initiated by typing
>@FILENAME

<EOF> is written at the end of the string, assuming no error is found.

It is shown in the Active Task list as ...AT.
(subsequent strings are called AT.Tn as in the TTY number)

3.8 USEFUL COMMANDS FOR FILE HANDLING.

LOG
Tells you which uic you are currently working under.
If it isn't [210,210], LOG 210,210 will change it.

DIR RAFIN.*
Gives directory of all files on DL0 with name RAFIN

TYP HKLZIG.DAT
Types the file HKLZIG.DAT

REN FOR002.DAT=BACKUP.BAK
Renames the file BACKUP.BAK to FOR002.DAT

COP RAFIN.DAT=FOR013.DAT
This creates a new file RAFIN.DAT, and puts a copy of FOR013.DAT in it. The file FOR013.DAT remains untouched.
It can also be used to join together files, e.g.
COP FOR004.DAT=FIRST.DAT,SECOND.DAT,LAST.DAT

CPN FOR002.DAT,FOR004.DAT
If a program is aborted whilst accessing another file, the other file is left ' LOCKED '. This command would unlock FOR002.DAT and FOR004.DAT.

PUR TOTO.*
This deletes all but the latest version of any file with name TOTO, whatever the extension.

WHE PARA.LSD
This searches through all possible uic's on disks DL0 and DL1 for versions of the file PARA.LSD. It tells you where it finds them.

DEL FOR004.DAT
This command deletes all files FOR004.DAT
WE RECOMMEND THAT YOU NEVER USE THIS COMMAND - IT IS DANGEROUS.

DMO DLn:/DEV
This logically dismounts the disk DLn (n = drive number) from the system (disk does not necessarily have to be running)

MOU DLn:/OVR or

MOU DLn:LABEL
This logically mounts the disk DLn: (n = drive number) on the system. LABEL is the logical name of the disk.
Disk MUST be running and READY and also this command must be preceded by a DMO, or horrible confusions may arise.

3.9 OTHER COMMANDS YOU MIGHT USE

TEC COLL5N.DAT
This opens the file COLL5N.DAT ready for editing using the system's editing routine TECO. See later chapter.

TEC <CR>
* EW LISHKL.DAT <ESC ESC>
This creates the file LISHKL.DAT and leaves it open for building and editing using TECO, see 11.4, p.11-6.

LSD
This calls the program that is used to control the diffractometer. A complete description follows in this manual, see chapter 4 and appendix A.

PCP
This is the command that is used to interrupt a counting sequence in the control program LSD. See p.4-2 and A-17.

KLS
This kills LSD in a hurry, should only be used in an emergency as PCP is normally more elegant but only works in a counting sequence.

KCO
This is the command that pauses COLL5N, the data reduction program, without leaving files locked. See p.8-13.

DISmount
This dismounts the data disk and gives information about what is written on it.

KDI
This cleans off the old data disk and prepares it for new data to be written.
Danger, scratched data is impossible to recover.

*** NOTE ***

Do not use DIS or KDI without reading the chapter 9.

BRO TTx: text
This sends the text to teletype TTx (x is normally 0, 1 or 3)

3.10 ACKNOWLEDGEMENTS

The direct command language demonstrated in this chapter was conceived and written, using existing DEC software, by A.Barthelemy, C.Turfat and P.G.Rice.

CHAPTER 4

THE LSD PROGRAM

(see also appendix A)

4.1 INTRODUCTION

The program LSD takes care of all handling of the diffractometer such as positionings, scans, reading of encoders, centrering of reflections and measurements. A number of commands are described below.

To run the program do
>LSD
The program gives time and date and then a '*'.
This is what we call the first level of LSD.

There are a large number of routines that you can call from this point. The program will give a response when it has done what you have asked it to.
DON'T TRY TO DO ANYTHING IN THE MEANTIME !!!

Some examples :
One routine is PAR [parameter section]. Type
PAR
The program responds in PAR with P. When parameters have been changed , finish with
END
The program comes back with the '*' of the first level.

At present the same applies to all commands involving the CAMAC (positionings, scans, etc.). To use these commands say
CAM
In the section CAM the program responds with a 'c'.
Return to main level using
END (as before).

4.2 MANUAL OPERATIONS OF THE DIFFRACTOMETER

The manual operation is very simple and an explanation will be given at the instrument.

4.3 EXITING FROM LSD.

When program is waiting for input with a '*',
normal exit from program is by typing
FIN
Avoid using ABO LSD if possible -
it may cause you to lose files.
Unfortunately, sometimes there is no other choice as PCP only works at specific times.

A smooth return to the waiting for input level (*) can be done during a scan by using the program PCP.

Run the program PCP by typing PCP and answer questions.
Normally the number 3 should be given.

4.4 COMMUNICATION WITH LSD

The rules for interaction with LSD are

** ONE **

All commands have three letters. These can be followed in some cases by numbers.
Example:

CEN0 1 1 -1
Centers reflection 1 1 -1 using the orientation matrix available in the program.

CEN4 1 1 -2
Centers reflection starting with the present diffractometer angles. The reflection will be called 1 1 -2.

Here CEN0 and CEN4 are particular options of the centering routine CEN, see p.A-12.

** TWO **

All numbers input are in free format.
The program divides the numbers up by looking for spaces.

** THREE **

If you want to change something, the command will be followed by ':'
Example:
POM: 52.05
Positions Omega at 52.05 degrees.

** FOUR **

If you want to read something, there is no ':'
The program gives the number(s) in question.
Example:
POM
Program gives the angle. (52.05, if the command follows the command under THREE)

** FIVE **

'=' is given if the number should not be changed.
Example:
ZER:= 0.51 -.1
Program keeps old value of 2Theta zero, but sets the zeroes of Omega and Chi to 0.51 and -0.10, respectively.

** SIX **

You only have to give the numbers necessary.
Example:
ZER: = .49
program keeps old values of zeroes for 2Theta, Chi and Phi.

4.5 SIMPLE DEMONSTRATION

The following gives a limited demonstration of the operations. Remarks are given in parentheses.

```
>LSD              (start program)
(the date is given)
*                 (the program answers , ready)
CAM               (calls camac routine )
C                 (responds with a 'C')
PPP               (ask for 2Theta, Omega, Chi, Phi)
(program gives angles)
C                 (C says its finished)
END               (finish with CAM, back to first level)
*                 (we are back in first level)
PAR               (we want to change parameters)
P                 (program answers)
MOU:1             (we want minimal output in scans)
P                 (ready for next command)
WAV               (read lambda)
(program gives lambda)
P                 (ready for next command)
END               (return from parameter change)
*                 (program answers with '*')
CAM               (back into CAM)
C                 (program gives 'C' )
PPP: 40 20.3 = -14.7 (Posn. 2Theta,Omega and Phi. Leave Chi)
C                 (says its finished, next command please.)
SOM:19 21.2 .2 10000 (scan omega from 19 to 21.2
                  in step of .2 with monitor 10000)
(program gives counts, but not angles)
C                 (program gives a 'C' when its finished)
END               (back to first level)
*                 (back at first level)
PAR               (so back to change parameter)
P
MOU:2             (we want angle and count)
P                 (done)
END
*                 (back to first level)
CAM               (back into cam)
C
SOM: 18.5 22.5 .2 10000 (redo scan but wider)
(program now gives angles and counts, line by line)
C                  (finished)
END                (back to first level)
*
CEN4 1 1 2        (i.e. This is the 1 1 2 refn and we center it)
(program gives results while centrering)
*                 (when finished gives '*' and waits)
FIN               (end of LSD)
(gives time and date and returns to the monitor)
```

Appendix A is a summary of the commands used to run LSD.

4.6 PARAME - LIST PARAMETERS

These program are available for listing the parameters,

PARAME will list all parameters which are being used in a specific type of measurement. Do

>RUN PARAME
The program responds ' GIVE A COMMAND '.
Give the LSD command corresponding to the measurement. The commands that can be used for PARAME are
CEN, MES, HKL, OPT, RST, QSC, REN, ROU, MUT, SEQ and ALL.

When its finished its output, give a carriage return to exit from program.

Useful at the end of a section of measurement so that you have a record of all the parameters you have used.

4.7 ACKNOWLEDGEMENTS

The LSD program has been written by A.Barthelemy, A.Filhol, P.G.Rice and C.Turfat in discussion with M.S.Lehmann, S.A.Mason, C.Zeyen and M.Thomas.

CHAPTER 5

THE INDEX AND RAFIN PROGRAMS

5.1 INDEX

The program indexes reflections on the basis of observed 2Theta, Omega, Chi, Phi angles when the cell constants and wavelength are known.
It does not take into account systematic extinctions.

The process, when successful, has three steps.

First, it calculates for each set of observations, all possible HKL's for which THETA(CALC) lies within THETA(OBS) +/- DELTA THETA.
DELTA THETA is given - see below -.
For DELTA THETA = 0, the value defaults to 0.05.

Second, it finds for all combinations of two sets of observations, the angle between the indexed HKL's for which ANGLE(CALC) lies within ANGLE(OBS) +/- DELTA.
DELTA given - see below -. Delta = 0 causes default to 0.2.

Finally, it finds all sets of indexed HKL's that explain all angles between the observed sets of Omega, Chi and Phi.

The user will normally be presented with several possible sets of HKL's which fit.
- they are already tested for right-handedness -
and he must now choose which set he likes the most.

If he wishes he may now specify which set of reflections he likes and the program will then set up the input file for the program RAFIN, - see next section -.
The program ensures that the first two reflections are acceptable to RAFIN. The user must say whether he wants the UB matrix written directly into LSD (for instant use) and which file he wants his RAFIN input to come from (usually RAFIN.DAT).
The program RAFIN can then be used (see 5.2).

Input to INDEX can be done either from the terminal or from a file INDEX.DAT
The format is the same, an example is given here.

```
THIS IS A DUMMY EXAMPLE          ( text )
5.82 16.15 4.09 90 103.2 90 .84 1.0
                                 (cell constants, lambda, DELTA)
16.18 9.01 34.71 14.74 0         (2Th, Om, Ch, Ph, DELTA THETA)
13.21 7.8 .71 -56.13 0.1         (2Th, Om, Ch, Ph, DELTA THETA)
etc. etc.
0                                ( end list with 2Th = 0 )
```

The program will only suggest sets of indexed HKL's if all reflections are explained. If not, the user must himself look through the list of observed and calculated angles to find a partial list.

5.2 RAFIN

Program determines orientation matrix (UB) from two or more sets of orientation angles for reflections, and refines (optional) wavelength; zeroes of 2Theta, Omega or Chi; a, b, c, alpha, beta or gamma.

To run the program do
>RUN RAFIN
after having set up the input file.

The input data are normally one of the following files:
RAFIN.DAT,FOR002.DAT,FOR003.DAT
(decwriter input can be used, but is cumbersome)

Use TECO to make or correct the file.

An example of the input file (comments in parentheses) :--

```
0 1 (second no.  0/1 for UB not transferred/transferred to LSD)
0 (always)
6 (number of reflections)
0 -4 -2 28.01 13.75 81.59 42.05 (H K L 2Theta Omega Chi Phi)
4 -6 7 50.84 25.37 34.04 18.41
-2 -6 0 41.55 20.53 66.93 59.99
4 0 4 19.74 9.94 -16.92 -5.40
1 -5 -3 35.59 17.70 82.32 1.40
6 0 0 18.47 9.26 -2.32 -46.95
0 .8405 (0/1 do not/do refine lambda; and lambda)
0 0.0 1 0.0 1 0.0 (0/1 for do not/do refine, 2theta zero,
                   0/1 for do not/do refine, Om zero,
                   0/1 for do not/do refine, Chi zero.)
0 15.9158 0 7.1939 0 14.277 0 90 0 98.72 0 90
                   (same for a, b, c, alpha, beta and gamma)
2 0 0 (H K L list for angles to be calculated)
0 2 0
0 0 2
0 0 0 (end of list)
-1 (end of input file)
```

Ensure that LSD is not running if you wish to transfer the matrix and wavelength directly into its parameter section, otherwise it may not be successful.
RAFIN will never modify the zeros for you. This is for you to do by adding them to the values in ZER of PAR.
REMEMBER THAT FOR A WELL ALIGNED DIFFRACTOMETER, THEY WILL NEVER CHANGE BY VERY MUCH.

Note- The first two reflections should be far away enough in reciprocal space to define a plane, At least 45 deg apart in Phi and only one may have Chi greater than 45 deg.

Note also that higher angle (theta) refins usually give a better fit.

You cannot, obviously, refine lambda and your cell at the same time.

5.3 ACKNOWLEDGEMENTS

The INDEX program has been written by M.S.Lehmann, and J.M.Savariault.

The RAFIN program has been written by A.Filhol and M.Thomas.
It has been implemented on the PDP 11 system by A.Barthelemy.

CHAPTER 6

THE HKLGEN PROGRAM

This program is used to generate a list of HKL's which can be used for input to the measurement routines of LSD.
LSD has a similar routine itself, although this is rarely used as it is considered easier to measure from a list.

To run the program do

>RUN HKLGEN
Input to the program is either from the terminal or from a file HKLGEN.DAT, already created by the user.

HKLGEN will generate HKL's according to min and max specified indices, and will write them into output file(s) if they are inside the Theta limits.

If CHI and PHI limits are specified, the program will also look to see if the HKL is measureable inside these machine limits.

If not, it will see if the Friedel Pair is inside limits

If this is also outside limits, it will see if the reflection can be measured for HKL PSI=180
-NOTE this option means CHI = 90-180 i.e. UP-SIDE-DOWN.

If measurement is not possible for any of these conditions, the HKL is declared BLIND.

Comments like FR.FR HICHI BLIND indicate these on TTY output.

6.1 INPUT FROM THE TERMINAL

The first question, under this option, that will be asked is whether a file HKLGEN.DAT should be created.
If subsequent runs are envisaged, this might be a good idea. In this case TECO can be used to make small changes to the input and the program can be quickly re-run.

HKLGEN then asks for a title - upto 80 characters to be displayed at the top of the output.

HKLGEN then asks for the wavelength and 9 extinction rules.
If you give wavelength = 0, the wavelength, extinction rules, and orientation matrix will be taken from LSD'S parameter files.
CHI and PHI software limits are also taken but an opportunity is given to over-write them.
If the wavelength is given explicitly, followed by up to 9 numbers for the ' rules limiting possible reflections ' (as in appendix-B), the orientation matrix must then be given line by line.

Theta, (NOTE NOT 2-Theta limits) are then asked for.
CHI and PHI limits (if required) must be given as well in this line.
If zeroes are given, no limits will be included in the calculations.
If nothing is given, LSD's limits will be used if data was taken from the parameter files, or it will default to zeroes if the data above was given by the user.

Three numbers are now requested for the relative speed of variation of HKL.
First number is the slowest changing index, third number is the fastest changing index.
1/2/3 is used to represent H/K/L
e.g. 3 2 1 means L changes slowest, then K, with H changing fastest.

Minimum and maximum indices in HKL are now requested.
You must give HMIN HMAX KMIN KMAX LMIN LMAX.
Note however that before starting the calculations, HKLGEN calculates itself what is the maximum value for each index for the specified Theta range and if this is inside these values, they will be modified.
Therefore, if, for example 0 999 0 999 -999 999 is given, HKLGEN will calculate the maximum values and give HKL's for positive H, positive K, positive and negative L.

HKLGEN then asks for four numbers :
The first NPUNCH concerns the HKL output file.
0 = no file for output
1 = file for output containing HKL only in 3I4 format.
2 = file for D15-REN (not used at RISOE)
3 = file for output containing HKL and setting angles.

The second IPARA
0 = Bissecting geometry - (normal).
1 = Normal beam geometry - (rarely used)
3 = D15 lifting counter mode (used with NPUNCH = 2)

The third number NBITES
1 = write HKL for each case in four separate files.
0 = write all HKL's in on file FOR00X.DAT
(X specified below).

The fourth number NLIST.
0 = write each HKL on TTY
(can take time and consume paper).
1 = suppress most of the output on TTY.

If in previous line FOR00X.DAT is specified for HKL output, X must be specified.
This is the last line in the input but need not always be given.

The program then normally generates as specified, creating file(s) if required. It gives a resume at the end and exits.

6.2 INPUT FROM FILE HKLGEN.DAT

Input is given in exactly the same order as above, so for a more detailed description of each parameter see previous section.

Two possible examples are given below.

```
KNUDDERKRYSTAL LAVET AF AARKVARD, 120K  (Text, 80 characters)
0.8405 0 0 0 0 0 0 0 0 0                (Wavelength and the 9
                                         Extinction rules. )
0.043361 -.04190 .5399
-.046409 -.032053 .03721          ( UB - three separate lines. )
-.00256 -.12861 -.02687
0. 36 -20 95                      ( Theta limits, Chi limits -
                                    note no limits on Phi. )
2 1 3                             (relative speeds of HKL.
                                    K slowest - L fastest. )
-99 -1 0 5 -99 99                (Hmin,Hmax,Kmin,etc.
                                  Note: H's are neg., K's is
                                  from 0 to 5, all L's - i.e.
                                  both pos. and neg.)
1 0 0 1                           ( Output file of HKL.
                                    Bisecting geom - usual-.
                                    All HKL's in FOR00X.DAT.
                                    Suppress TTY output. )
3                                 ( HKL file on FOR003.DAT )
```

Example two - probably more normal :--

```
SAMPLE B SHELL 30-35 DEG                ( Title )
0                          ( Lambda UB, Extinction rules,
                            PHI and CHI limits taken from LSD )
30 35                      ( Theta limits; note CHI
                            and PHI limits are OK )
1 2 3                      ( H slowest - L fastest )
-99 0 -99 99 0 99          ( all neg H - all K - all pos L )
1 0 1 0                    ( output file of HKL only.
                            Bisecting - usual.
                            Four files of NORMAL.HKL.
                                          FRPRS.HKL.
                                          HICHI.HKL.
                                          BLIND.HKL.
                            List all on TTY.  )
```

6.3 ACKNOWLEDGEMENTS

HKLGEN is a program which has evolved in the hands of :
A. Filhol, S. Mason. A. Barthelemy and J. Allibon.

CHAPTER 7

THE LISCAR AND DATFIR PROGRAMS

7.1 LISCAR

This program is used to give information about the state of data disks.
It reads from an assignable device TR1:

To look at a data disk mounted on DL1: do
>ASN DL1:=TR1:/GBL

The assigned device TR1: should be left assigned to DL1: (assuming that LSD is writing to DL1:)

Then do

>RUN LISCAR

If the program gives an error, it will mean normally that either
The disk is not properly mounted,
or you are under the wrong [uic],
or you went wrong in the assignment.

7.2 DATFIR

This program works in the same way as LISCAR except that it is used to examine the data taken for specific reflection numbers. Do

RUN DATFIR

and it will ask you for the first and last numbers of the reflections you wish to examine. Don't ask for more than is necessary, as a lot of garbage is outputted.

There then follows a complete list of all the numbers stored for the reflections asked for.

This program should not be needed in the average experiment.

7.3 ACKNOWLEDGEMENTS

The LISCAR and DATFIR programs have been written for the PDP 11 system by A.Barthelemy and C.Turfat.

CHAPTER 8

THE COLL5N DATA REDUCTION ROUTINES.

8.1 COLL5N ROUTINES

The COLL5N routines do on-line/off-line data reduction.
They provide the user with any of the following:

Several choices of schematic representation of the data on the terminal.

Two files, written on the relevant data disk, containing information, scan by scan, on the measurement.
When collecting normal crystallographic structure data (not Q-scans), the first, the LOKAL file, contains one line of data per reflection with headings and a resume every 25. This is useful as a rapid reference for information on the data collected (structure factors, relevant angles etc.).
For Q-scans the LOKAL file is meaningless but C5GET.OFF contains the counts and all necessary information to reproduce the scan in the PUNCH file using PUNCH option 6 (see 8.4.2).
The second, the PUNCH file, contains as much data as you wish, and is the file that you would take away with you.

Absorption correction can be done on the data, assuming crystal details are known, directly in the COLL5N routines. The program used is a modified form of the Coppens DATAP routine.

Statistics on the data can be kept and updated as COLL5N runs. They provide information on the measurement such as FSQ/sig(FSQ), scan widths, peak centering etc.

COLL5N updates it's own personal file C5GET.OFF to record details of it's parameters and it's state relative to that of the measurement routine LSD. Using this, COLL5N can be paused and restarted without missing data. Similarly, should the program exit abnormally, it will be automatically re-started up to five times by a monitoring task C5VRFY.

Reduced data files, written on the data disks, should therefore be complete, and are not fragmented by program pauses.

8.2 STARTING THE ROUTINES

The system is conceived for running on two terminals, so run COLL5N on the terminal not used by LSD. Both programs can however be run on one terminal with a slightly confused output. One way of reducing the output is to use ISOR=0 (MOU in PAR) and DISPLAY option 4 or 0.

Type RUN COLL5N

The program responds with INITIALISE, CHANGE or RESTART.

Note that COLL5N generally only looks at the first two letters of a word. Any additional letters will be ignored up to the next space.

INitialise resets all parameters to zero - apart from those collected in STATS which must be re-set seperately - and forces you to pass through all the setup routines.
This should rarely be necessary, but should the file C5GET.OFF be lost or corrupted, you must initialise in order to reconstruct it.

CHange reads the file and gives you the possibility to modify parameters before starting the reduction routines. It tells you what the present value is before asking for the new value.

REstart reads the file and reconstitutes COLL5N from just before the pause. The treatment will then continue as if the pause had never happened.
After certain serious (rare) errors, REstart may be disabled. Here it will be necessary to explain the error and redefine relevant parameters before continuing.
A successfully completed run in non-auto mode will also disable REstart, which would after all be meaningless.

To stop when COLL5N is running, see 8.8

8.3 INFO AND RINFO, PARAMETERS FOR REDUCE

INitialise forces you to define these parameters.
CHange asks you if you want to change them, and shows you their present values before asking for the new values.

Most of these parameters can be left as their default values, also specified.

A detailed description of their functions is available at the diffractometer, a brief summary follows:

INFO(1) - first line -- 1st. number.
BETA The number of points to be taken as ' search background '.
Default is 10% of points.
If this is made larger than total number of points, different fixed backgrounds on each side can be specified in 6th and 7th numbers of this line.

INFO(2) - First line -- 2nd number.
K, significance level in tests on quality of data.
Default is 5.

INFO(3) - First line -- 3rd number.
Number of points to be added to the peak on either side.
Default is 1.

INFO(4) - First line -- 4th number.
M, fixed background for FSQ < M*SIGMA(FSQ)
Default is 3.

INFO(5) - First line -- 5th number.
Choice of diffractometer.
Default is 4 circle.

INFO(6),(7) - First line -- 6th and 7th numbers.
If BETA is larger than total number of points, these numbers are left background and right background, respectively.

RINFO(1) - Second line -- 1st number
Monitor count used to scale data.
Normally this would be made the same as in MPA in LSD, and kept fixed for the experiment.
Default is 100000

RINFO(2) - Second line -- 2nd number.
F, the profile shape factor for peak width correction.
Default is 2.

RINFO(3) - Second line -- 3rd number
B, part of profile which is used as background for INFO(4) above.
Default is 0.3

RINFO(4) - Second line -- 4th number.
Dead time in seconds for detector.
Default is 0.000007.

RINFO(5) - Second line -- 5th number.
Monitor counts/s. Used in dead-time correction on monitor.
Should not be given if reactor is stable.
Default is 0.

RINFO(6) - Second line -- 6th number.
Monitor dead-time in seconds.
Default is 0.000015

RINFO(7) - Second line -- 7th number.
Sum of (MU*T)'s for cryostat, (cylinders parallel to PHI)
for angles = 0
Default is 0 (transmission = 100%)

8.4 COLLSN OPTIONS

The COLLSN options are defined from the prompt

[***] Hit me ! :--

[***] shows whether you are initialising or changing .
The program expects one of the following commands

DISplay - to set up the terminal display option.

PUNch - to set up the output file option.

STAts - to setup the STATS option.

DATap - to set up the DATAP option.

QSC - to change defaults for the Q-scan display routine.

GO - to start the data treatment.

QUit - to terminate the program, updates the D-A file.

Anything else typed will respond with a help message.

The program will, to some extent, remember relevant parts of the input. An example: If you type DIS for the display routine, it will ask you what option you need and offer you a further help message if necessary.
If, on the other hand, you know that it is display option 4 that you want, typing

DIS 4

will be immediately taken by the program and the ' Hit me ' prompt will be re-shown.

8.4.1 THE DISPLAY OPTION.

There are at present five different types of visual display of the standard scan data to be shown on the terminal.

FULL OUTPUT means a detailed plot per reflection with a maximum of infotmation given after the 'plot', (angles along short edge of the paper, counts along long edge of the paper).

ONE LINE per reflection means just that. A minimum of information is given on one line and no visual plot is done.

VERTICAL COUNTS display gives some information on the data plus a schematic representation of the peak as follows:

```
                        1 1 2 1 1
                    1 5 8 0 9 2 8 4 2 1
          2 3 5 6 9 8 6 2 9 6 0 5 6 1 5 8 6 4 2 1
  5 9 3 7 5 7 9 0 5 2 5 7 6 1 0 9 6 0 6 5 7 0 8 2 7
                          c o
  0 0 0 0 * * * * * * * * * * * * * * * * 0 0 0 0 0 0
```

The actual count for each point reads vertically.
The line containing 'c' and 'o' shows the calculated and observed centres of the scan, If they coincide, a '*' is shown.
The last line shows how COLL5N treats the scan
Counts labelled '*' are peak.
Counts labelled '0' are background.
Counts labelled '-' are discarded.

The ' FIVE LINES ' option is similar to the previous, except that it tries to give the form of the reflection as follows:

```
                          * *
                        *     * *
                      *
                  * * * *             * * *   *
    0 0 0 0 0 *                               *   * 0 0 0 0 0
                          c o
      9   6  10  12  15     1026       16  17  10   9   6
```

Here, the 'height' of the scan is divided into 5 levels. The character shown for the point has the same designation as above, except that the maximum count is shown as '*'.
The line containing 'c' and 'o' is as above.
The last line shows the value of the max count in the centre.
Note- Not necessarily the count at the middle of the scan.
On either side of this it shows as many background points as it can in the space available. The format used is 1X,I3 so if a background count exceeds 999 it will scale the line to make this point equal 999 and a message is given.

Display options are:

```
  -1 = no display on terminal
   0 = full print              - one page per refln. ( default )
   1 = short print             - one line per refln.
   2 = short print except for error and test reflns
                               - gives full print.
   4 = ' vertical counts '     - about ten lines per refln.
   5 = ' five lines '          - about ten lines per refln.
```

For every n.th reflection to be printed only, add (n*10) to any of the above options, example, 54 says display only every fifth reflection with option 4.
Note however that error and test reflections will always be displayed.

8.4.2 THE PUNCH OPTION

COLL5N will write two files of reduced data on the data disk. These will be initialised at the start, for each new data disk and at certain parameter changes. Otherwise they will be appended to, and assuming that the measurement is running smoothly, they will be complete for each data disk. If COLL5N is stopped and restarted, the files will not be fragmented.

The LOKAL file contains one line of data per reflection with headings and resume of parameters every 25 reflns.
It is very useful as a quick reference sheet for the measurement.
To type it, stop COLL5N with KCO and QUit and then do
TYP DI1:LOKAL.COL - for automatic treatment.
TYP DR1:LOKAL.COL - for non-auto treatment.
Don't forget to REstart the treatment afterwards.

The PUNCH file can contain varying amounts of data per reflection. It is the reduced data that you would take away with your transfer to another computer via the modem lines.
It's name can be re-specified by the used, default is PUNCH.COL.

Punch options are as follows:

```
-1 = no files written
 0 = just LOKAL.COL file     - default
 1 = LOKAL.COL plus PUNCH file containing 1 line  per refln.
 3 = LOKAL.COL plus PUNCH file containing 2 lines per refln.
 4 = LOKAL.COL plus PUNCH file containing 2 lines per refln
                                    plus raw data counts.
 5 = LOKAL.COL plus PUNCH file containing 2 lines per refln
                                    plus raw data counts
                        plus their corresponding theta angles.
6 = PUNCH.COL file for Q-scan containing
```

a listing of h,k,l, reciprocal lattice vector (tau) in AA-1 (2PHI/d), count.

Each scan is processed separately from the screen terminal. If PUNCH option = 6 is used, the data will be prepared as input for the Algol programs QSCANPLOT or QSCANFIT under usercode NEUTRON on B7800.

1st line: FF
2nd line: scan number
3rd line: Date - to run a program on B7800, the format should be changed to be date, month, year, (e.g.: 18,4,1982)
4th line: Title text
5th line: No of points in the scan, 1, 1, 0.1, 3
The first three variables should be left unchanged.
The fourth variable corresponds to the interval-length (0.1) of the independent variable plotted as 5 mm along the abscissa.
The fifth variable indicates which one of the Miller indices variable (h(1), k(2), l(3), reciprocal lattice vector (4)) is plotted as the independent variable along the abscissa.
6th and following lines: h,k,l,tau,count.

The DATAP option, if engaged, gives an additional line after the second line.

The first line of the PUNCH file contains:
Numor, HKL, Fsq, sig(Fsq), Th, Om, Chi, Phi, Date,
in the format:
I6,3I4,2F10.2,4F8.2,10A1.

The second line, if requested, contains:
Numor, L and R bkgds, Peak to Bkgd ratio, Time per step, Monitor count, Points in L bkgd, Peak, R bkgd, FWHH, FWBACK, Offset from centre, Range in Omega and Theta, Steps in Omega and Theta, in the format:
I6,2I5,2F7.2,I5,2X,3I2,3F5.2,2F6.2,2F5.2.

The lines containing the counts, if requested, contain:
Numor, Line number and ten counts per line,
in the format:
I6,I2,10F7.0

The lines containing the angles, if requested, contain:
Numor, Line number and ten angles (theta) per line,
in the format:
I6,I2,10F7.2

WARNING:
Beware of the message ERROR WRITING PUNCHFILE which will create a new file called LOKAL.COL;2 and assign the name LOKAL.COL;1 to the original file.
To play safe do DIR DL1:*.COL to get a directory at the finish of the data treatment.

8.4.3 STATS OPTION

COLL5N can keep statistics on the data that it is treating. These are updated as COLL5N runs and are kept in a direct-access file STAT.DAT on DL0:
They are displayed either every n reflections, if n is defined, or, by default, at every group of test reflns.
The data file should only be set to zero at the beginning of a measurement. At all other times, COLL5N will take the old version and update that. Statistics are done on the following:

Fsq/sig(Fsq) :-- You define N ranges of Fsq/sig(Fsq).
(def 10, width 5 sig).
It displays the number of reflns in each range.
The number of negative intensities.
The number of reflns which are greater than max. range.
The number of reflns where Fsq > 3 sig(Fsq).

Data as a function of Theta :--
You define N ranges in Theta (default 10, width 6 deg)
COLL5N then tells you the number of reflns in each range.
Mean Fsq and mean sig (Fsq) for each range.
No of significant reflns (default >5 sigma) in each range.
Mean FWHH and FWBACK for each range (signif. reflns only).

Centering analysis:
You define N ranges in Theta (default 5, width 12 deg).
COLL5N then tells you the number of reflns in each range which are >-2, -1, 0, +1, >+2, steps off (signif. reflns only).

Approximate R factor
COLL5N calculates (SUM (sig(Fsq)) / SUM (Fsq)) for all data collected.

8.4.4 Q-SCAN DISPLAY OPTIONS

COLL5N will give a visual representation of Q-scan data on the teletype.
The plot given is similar to a mixture of both the ' Five Lines ' and ' Vertical Counts ' display options.
You may, if you wish, change the default for the number of lines in the display. Default is 10, Max is 20.
To create the PUNCH file correctly use PUNCH option 6.

In non-automatic mode, an option can be engaged, where COLL5N will treat only Q-scan data. Here, at the end of each treatment, the user will be asked if he wishes to redefine the maximum in the display. If he answers yes, the program will demand the new maximum and re-do the scan. It will then ask again if he wishes to redisplay the scan.

If he answers no, the program will ask for the next data-set number to be treated, which it will treat for you.

This option is terminated by asking for data-set -1 and the routine gives the prompt
COLL5N PAUSE, give QUIT, CHANGE or RESTART:--

NOTE

PUNCH option 6 has been implemented for QSCANS by Bente Lebech, July 1982. Modifications may still be necessary and will be implemented at the users requests.

8.4.5 DATAP, ABSORPTION CORRECTION PROGRAM.

DATAP, a modified form of a program obtained from S.U.N.Y (P.Coppens), can be run either seperately, on previously treated data, or as part of the COLL5N data treatment.

DATAP as part of the COLL5N routine.

Parameters are set up inside the COLL5N routine.

COLL5N will request the following for DATAP.

1. The wavelength, orientation matrix, and cell parameters. If it feels that these have already been defined, it will propose the present values and ask if the user wishes to change them.
   The user defines them by one of three ways :
   a. Entering them via the keyboard.
   b. Reading them from LSD's parameter file PARA.LSD.
   c. Reading them from a data file, in this case the data-set number must be given.

2. The crystal must then be defined.
   The number of faces is requested, followed by
   either - The indices of each face with the perpendicular distance from a common point inside the crystal boundaries (distance in cm).
   or - The machine setting angles for each face (in the diffracting position) with the perpendicular distance as above.
3. Then Mu, the absorption coefficient for the crystal must be given (units cm-1).
4. Finally the number of grid points along a, b, c, to define the Gaussian grid for integration are requested. The overall maximum is 1000, so for example 10 x 10 x 10 is the limit, (or 8 x 10 x 12). These must be even integers.

These parameters are then passed to DATAP which will set up the grid matrix before allowing COLL5N to continue.

During each data treatment, DATAP will be passed the HKL, and Fsq from which it will return, to COLL5N, the pathlengths, Tbar and corrected intensity, which are displayed as an extra line in the display option, and a supplementary line in the PUNCH file.

DATAP will be terminated with COLL5N.

DATAP as a stand-alone program.

DATAP can equally be requested to treat data which has already been reduced by COLL5N.
In this case, the parameters are passed to DATAP via a file DATAP.DAT which has the following format:

```
line 1  Title
line 2a Cell   ( if first number found is zero, cell, UB matrix
                 and wavelength are taken from PARA.LSD
                 and DATAP goes straight to line 3)
line 2b Orientation matrix  ( first  line )
line 2c    --*--     -*-    ( second line )
line 2d    --*--     -*-    ( third  line )
line 2e Wavelength

line 3  Grid points along a, b, c  ( max = 1000 total )

line 4a No. of faces and face defining option :
              ( option = 1 means define face by HKL
                option = 2 means define face by setting angles )
line 4b Indices for face 1 and perp. dist. from common point.
line 4c   -*-   -*-  -*- 2 -*-  -*-   -*-
line 4d   etc.

line 5  Mu (linear absorption for the sample crystal).
```

The FULL specification for the input file must be given (example DL1:JAKH3.COL). This will be a PUNCH file already written by COLL5N.
The output file will be identical, except that it will contain one supplementary line per refln. containing the DATAP results. If the input file already contained the 'DATAP' line, this will be overwritten by the new results in the output file.

Before terminating, DATAP will ask if you have more reduced data files to treat.

8.4.6 THE QUIT COMMAND

QUit causes COLL5N to exit. It has the same effect whether typed the prompt ' Hit Me ! :-- ' or
COLL5N PAUSE, give QUit, CHange or REstart :--

Before terminating, a brief resume of the run that is being stopped/paused will be given,
The file C5GET.OFF will be updated.
Tasks C5VRFY and DATAP, if running, will be terminated.
COLL5N will then say if a restart is possible, exit and return you to the monitor.

8.4.7 THE GO COMMAND.

If the treatment is being initialised, COLL5N will ask if an automatic data reduction is wanted.
If you say Yes, assuming no reflections have been missed, data will be treated as soon as LSD has finished writing it.
Data will be taken from dummy device DI1:
If you answer No, it will ask for the first and the last number to be treated, and look for this data on dummy device DR1:

If you have just finished a parameter change and the previous state was obviously uncompleted, COLL5N will remind you of what you were doing and ask you if you wish to carry on.

8.5 DATA REDUCTION

8.5.1 AUTOMATIC DATA REDUCTION

As we have already stated, automatic data reduction means that COLL5N will read untreated data files from and write treated data files to the dummy device DI1:
This is where LSD will be writing its data files.

Before starting the treatment, you are asked whether you wish to catch up some missed refins before going automatic or not. This is necessary if COLL5N has been started some refins after LSD has begun writing. These refins will not be treated as COLL5N has not yet ' learned ' the state of the measurement.

Once the treatment has started, COLL5N knows the state of the data disk and will check, just before treating each refin, that the data disk has not changed, or that no data has been missed etc etc

If it finds that the data disk is empty, ie a new data disk, it will give a message and re-create the data-files as necessary. For this, when restarting after having changed the disk, it is preferable to restart COLL5N before restarting LSD.

8.5.2 NON-AUTOMATIC DATA REDUCTION

Here, data is read from and written to dummy device DR1:

COLL5N will ask for the first and the last number and, assuming no serious errors are found, it will treat these data-sets and then exit.

(Don't forget that if you are retreating previously treated data, you may be taking statistics on the same data twice.)

In both cases KCO will cause COLL5N to pause from which point the treatment can be stopped, changed or restarted.

8.6 THE PROGRAM C5NCHK.

If the measurement is running smoothly, you will have on each data disk a complete LOKAL file, useful for a quick reference on that disk's data, and a PUNCH file.

The program C5NCHK can be used to verify the completeness of your PUNCH files, and, if necessary, to amend any small problems with them.
Firstly, you must see how many PUNCH files you have on the data disk.
DIR DLn:PUNCH.COL;* will show you all versions of this file on the disk; n is the number of the disk drive that holds the data disk.
Under normal circumstances there will only be one PUNCH file, but should there be several, C5NCHK will give you the opportunity to write them together.

The program will also, in a way similar to COLL5N, do statistics on the data that it finds in the PUNCH file.
If requested, it will create a file on the data disk containing statistics on that disk only. It will also offer the possibility of creating and updating a statistics file on DL0: containing details of several disks, this would be useful if you should lose your COLL5N statistics.

RUN C5NCHK call the program.

It asks you for the number of the disk drive that holds your data disk, normally 1
It asks you for the PUNCH file-name on that disk. If you are treating one of several files on the same disk, you must give the version number.
It will ask you if you wish to re-write the file(s). This would be useful to add together several PUNCH files produced inadvertantly on the data disk. It could also be used to copy the file(s) onto DL0: for example.
If you say Yes, you are asked for the output file name and whether it already exists, in which case it will be appended to.
It asks you if you want to do statistics on the data file.
If you say Yes, you will have, when the program has terminated, a file on the data disk which contains statistics on your PUNCH file similar to the COLL5N on-line option.

You can also do statistics in a similar way on several PUNCH files. If you say Yes to this, it will ask you if you are continuing a series of runs, in which case this disks data statistics will be incorporated into it. The file for this is written on DK0:.
Both files have the PUNCH file-name with the extension 'STA'.

C5NCHK will then run through the file and inform you of any anomalies in the sequence of numor.
When it finds the end of file, it will display the statistics collected on the data read (if requested), and then ask you if you want to carry on.
If you say Yes, it will ask you if you are treating a new disk, or continuing with the same. If you change disk, it will re-create the statistics file and/or the output file as necessary.

8.7 THE PROGRAM STA.

This displays the contents of a STATS direct-access file whether written by COLL5N or C5NCHK.

STA call the program

It asks for the name of the file that you want to read, example DL1:JAKH.STA
If you give <CR> it will look for the current STAT.DAT on DL0:

8.8 HOW TO STOP COLL5N

Type KC0, the computer answers

COLL5N PAUSE, give QUit, CHange, REstart :--
Then type 'QU' and the computer answers

REDUCTION STOPPED BY USER

8.9 ACKNOWLEDGEMENTS

The COLL5N data reduction program was written originally by M.S.Lehmann and S.Wilson.
Then J.Allibon got his grubby little hands on it and has made it so complicated that no-one is ever going to dare to use it again.

The COLL5I interactive data reduction program has been written by A.Filhole, M.S.Lehmann, C.Turfat helped by A.Barthelemy.

CHAPTER 9

CHANGE OF DATA- AND SYSTEM DISK

9.1 CHANGE OF DATA DISK

For each scan measured, LSD creates on dummy device DI1: (normally assigned to DL1:) one complete data file named, for example R20362.DAT;1. The number 20362 corresponds to the data set number shown on the LSD printout and is in the range 1 to 30000. Note that the file created is always version No. 1. This means that if the data set number is for any reason reset in LSD, there is a large risk that data will be overwritten.
The data disk will hold about 2500 such data files. When it is full LSD will give an error message and pause.
To avoid this, please feel free to stop the measurement and change the data disk at any convenient time.
A warning is given by LSD at the end of the 900th, and all subsequent data files written.

Normal running, as already stated, is:--

```
Writing data to DL1:            ( -- DL1:=DI1: )
Transferring data from DL1:     ( -- DL1:=TR1: )
```

To change data disk, here DL1: :--

First, Stop LSD.
Bring LSD to it's first level, '*', with the program PCP.
Type PCP <CR> (on any terminal)
And answer the question with '3'
Then stop LSD by typing FIN.

NOTE

PCP = 1, (goes to LSD PAUSE), will normally NOT be sufficient to allow you to dismount the data disk. LSD is still running and accessing files on DI1:. Should however, LSD have an error on

the data disk, for example, a full disk, it will give an error message and, similarly, go to LSD PAUSE. You are here able to dismount the disk.
(carry on with a clean disk with RES LSD.)

Then stop COLL5N:
Bring COLL5N to it's first level by typing KCO.
Then exit by typing QUit.

Ensure that there are no other user tasks active on the system.
Examples, T11V10, DESIRE, RAFIN, etc.

Type ACT /ALL (answers for example :--)
...LDR
...MCR
MCR...
F11ACP
SETTCU

Do not try to abort system tasks.
Examples, ...LDR ...MCR F11ACP SETTCU ...SPY

You are now ready to dismount the data disk.

Type DMO DL1:/DEV
The computer answers:
DMO -- TT0:DISMOUNTED FROM DL1: *** FINAL DISMOUNT ***
*** DL1: -- DISMOUNT COMPLETE

Stop the disk by releasing the yellow 'LOAD' switch; the white 'READY' light thereby goes out. Wait for the yellow 'LOAD' light to come on. Pull out the disk drive and open the drive from the top. Remove first the disk top-cover and then the disk by lifting it in the handle while the switch on the handle is pressed to the left. When out, reassemble disk and cover.
Take the new disk and lift it from the top-cover with the handle while the switch on the handle is pressed to the left. Place the disk in the drive and lay down the handle. Put the top-cover over the disk and close the drive. Press the 'LOAD' switch, the yellow 'LOAD' listh should go out. (If it does not the disk is probably not correctly placed in the drive or you have forsotten the top-cover). Wait for the 'READY' light to come on, then type

MOU DL1:/OVR
The computer answers:
>
This initiates a command string that initialise the new data disk.

Restart COLL5N with RUN COLL5N and REstart. It should recognise that it has an emty data disc and re-initialise the reduced

data files.

Restart LSD with RST.

9.1.1 SUMMARY FOR NORMAL ACTION

1. Stop LSD with PCP = 3
   and FIN

2. Stop COLL5N with KCO and QUit.

3. Type ACT /ALL to check that no user tasks are active.

4. Type DMO DL1:/DEV

5. Stop and remove disk.
   Put in the next data disk in the sequence,
   start drive and wait until drive is READY.

6. Type MOU DL1:/OVR

7. Restart COLL5N with REstart.

8. Restart LSD with RST.

9.2 CHANGE OF SYSTEM DISK

9.2.1 SUMMARY FOR NORMAL ACTION

1. As 9.1.1

2. As 9.1.2

3. As 9.1.3

4. Stop and remove disk.
   Put in the other system disk,
   start drive and wait until drive is READY

5. Bootstrap the system as in 13.1.1 and 13.1.2

CHAPTER 10

THE DISPLAY, VERIFY, SETPOS AND POSNUL PROGRAMS

10.1 THE DISPLAY PROGRAM

The display program DISPLAY is installed by the startup command file STARTUP.CMD, during start of the system. The display program is normally called by the LSD program, but it can also be called by the operator by typing

```
>RUN DISPLA
```

After start the display program reads the file

DL:[210,210]BUTEES.LSD;1

which contains the software offsets of the 4 axis. Hereafter the program enters an infinitiv loop doing:

1) Suspend program execution for 100 mS
2) Read display SWITCH
3) Go to read sequence for device selected by display switch. Devices are:
   - 1 = PHI Encoder
   - 2 = CHI Encoder
   - 3 = OMEGA Encoder
   - 4 = 2TH Encoder
   - 5 = P-AXIS Encoder
   - 6 = Not used
   - 7 = Not used
   - 8 = Scaler 1
   - 9 = Scaler 2
   - 10 = Scaler 3
   - 11 = Scaler preset
   - 12 = Not used
4) Read selected device. If PHI, CHI, OMEGA or 2THETA encoder is selected, then add offset value.
5) Display value
6) Go to no.1

It should be noted, that the software offsets is only read ones, and not for each display refresh.

Therefore it is possibel to change the contents of the offset file while DISPLAY is running, whithout affecting the display reading.
DISPLAY must be aborted and restarted again before the change is displayed.

10.2 THE VERIF PROGRAM

10.2.1 INTRODUCTION

The VERIF program verifies, that the positions of the axis are correct.
If there is a small deviation, the program will make a correction.
With a bis deviation the program can not make a correction, and it will be nescessary to establish a correction by manual means.

10.2.2 USING THE VERIF PROGRAM

To run the VERIF program type

```
>RUN VERIF
```

then the axis will start running. When the axis stops, VERIF will give a message.
If all the axis has been on a correct position, VERIF writes

```
TTO - - - STOP
>
```

If there has been a small deviation (less than 1.20 degree) on one or more of the axis, VERIF writes e.g.

```
2TH= 0 OMEGA= -38 CHI= -90 PHI= 0
```

A small error, less than 1.20 degrees deviation, is automatically corrected, when the VERIF program is run, so both OMEGA and CHI are now on the correct position.
If there has been a deviation on more than 1.20 degree, VERIF writes e.g.

2TH= 0 OMEGA= 120 CHI= -90 PHI= 0

In this case no correction takes place for OMEGA, and a correction of the position must be established by manual means of B.LEBECH, F.KREBS LARSEN or M.H.NIELSEN.

10.2.3 DESCRIPTION OF THE VERIFY PROGRAM

The verify program VERIF verifies, that the zero reading of the incremental position encoders coincides with the appearence of the zero position indicator pulses from the optical zero point detectors on the four axis.

The incremental position encoders are counters integrated in the CAMAC stepping motor controller modules (P1082). These counters counts the steps issued to the stepping motors.
Up, for positiv direction and down for negativ direction, thus keeping track of the motor positions at all times.

As the optical zero point detector pulses only exists during a travel of 1/100 degree, it is convenient to detect them and record theire positions, while the axis are on the move. This is carried out in the CAMAC zero point detector module P1304.

The positions of the zero point pulses are recorded by counters starting to count motor steps (in positiv direction only), when the corresponding zero point detector pulse appears.
The motor should be stopped before the maximum number of steps, which is 240 (= 1.20 degree), has been accumulated. If it is not stopped, the counter will stay at the maximum count which is then interpreted as 'overflow' by the VERIF program.

Now, VERIF works as follows: First the current four positions (incremental encoders) are recorded. Then all axis are returned to -2.60 degree. The axis may or may not have passed the zero point.

Next, all axis are run 2 degrees forward, setting all positions to -0.6 degrees. This assures, that an axis having passed zero in the first operation will have reached overflow and therefore has disabled counting until a new zero position pulse has been received. Then the four zero point detector counters are cleared to zero.

Next, all axis are run 1.2 degree forward leaving them at what is expected to be +0.6 degree. The zero point detectors are then read, and it is checked, that the reading is 120 steps = 0.6 degree.

If it is not 0.6 degree, it is checked whether an overflow has occured. If there was no overflow, the incremental encoder setting is corrected corresponding to the zero point detector reading. Thus, small errors, less than 120 steps deviation, are automatically corrected,when the VERIF program is run.

The HUBER mechanics does not make it possible to position the optical zero point detectors exactly at the nonius zero point. Neither does it make it possible to mount the optical zero point detectors exactly at the same positions after removal.
To make nonius zero coincide with detector zero, VERIF therefore contains four offsets, one for each axis, which are the distances in steps from nonius zero to detector zero.
They appear in the FORTRAN program as D2TH, DOMEGA, DCHI and DPHI, and their values must be changed whenever the detectors are removed and mounted again.

10.3 THE POSNUL PROGRAM

The program POSNUL runs all four axis to the nonius zero positions. This makes it easy to verify that no steps has been lost.
POSNUL is mostly used during lineup of the diffractometer after the zero point detectors have been removed and mounted again.
To invoke POSNUL type

```
>RUN POSNUL
```

10.4 THE SETPOS PROGRAM

The routine is used to set wanted numbers into the position counters in the four CAMAC motor control modules. These counters will normally hold the position of the four axis but e.g. after power has been switched off their contents are lost and new values must be read in with the SETPOS routine. Type on the terminal

```
>RUN SETPOS
```

The terminal will print

```
INPUT POSITION P1,P2,P3,P4:
```

Now type the position of the four axis. Watch on the display that the four values typed are set correctly.

10.5 ACKNOWLEDGEMENTS

The DISPLAY, VERIFY, POSNUL and SETPOS programs have been written by J.Bundgaard and P.Skaarup.

CHAPTER 11

THE TEXT CORRECTION PROGRAM TECO

11.1 INTRODUCTION

TECO is a system program. TECO can be used on any existing ASCII file.
TECO uses a buffer which contains up to one page of characters of your file. This is quite a large unit and most normal files will not fill it. Therefore you will probably not need to cross pages. If you do cross a page (see 11.3.1 p.11-2), you can no longer edit on the previous page without exiting from TECO and starting over again.
The program is at any time pointing at some position in the buffer. At the start the pointer is before the first character.

11.2 CALLING TECO

To edit a file, the full command for calling TECO would be for example :

```
TEC DEV:FILENM.EXT;VERSION NO.
```

At exit of TECO a new file is created with the same name but a higher version number.

Example: Followings files of type RAFIN.DAT are present on DK0:
RAFIN.DAT;236
RAFIN.DAT;237
RAFIN.DAT;240

By using the command
>TEC RAFIN.DAT
the file RAFIN.DAT;240 is the source. At exit RAFIN.DAT;241 is the corrected file. RAFIN.DAT;240 is, in principle, unchanged.

TECO responds with a *.

Commands are separated, where necessary, with $ (ESC or ALT on keyboard, known as an ALTMODE).

Two consecutive $'s cause TECO to act on the command string.

Note therefore that commands are no longer terminated with <CR>.

Commands can be compounded, TECO acts on them in the order that it finds them.

11.3 SIMPLE TECO COMMANDS

A TECO manual is available at the machine containing all TECO commands. Hopefully though, the following, short description will demonstrate all the commands that you will need.

11.3.1 POINTER MANIPULATION COMMANDS

J

Positions the pointer back at the beginning of the buffer, i.e. in front of the first character.

nC

Moves the pointer across n characters.
If n is neg. moves the pointer backwards along the line.
If n is pos. moves pointer forwards along the line.
If n is not given, it moves pointer forward one character.

NOTE : Carriage return is in fact TWO characters.
CARRIAGE RETURN + LINE FEED.

nL

Moves pointer n lines and positions at the beginning of that line.
If n is neg. moves back n lines.
If n = 0 moves pointer back to beginning of current line.
If n is pos. moves forward n lines.

P

Writes the content of the buffer onto the output file, then clears the buffer and reads the next page of the input file into the buffer. The pointer is left in front of the first character of the new page.

11.3.2 TEXT TYPE-OUT COMMANDS

nT

Type n lines (n pos. or neg.). Pointer does not move.
If n is omitted, types from pointer to and including next line feed character.
If n is neg. types the n lines preceding the current line.
If n = 0, i.e. 0T, types from beginning of line up to the pointer.

HT

Types out the entire content of the buffer. If you have a big file, this is clumsy and might take a while but for small files it is useful. The pointer position does not change.

V

Types out the current line without moving the pointer. Equivalent to 0TT. Take care you know where the pointer is.

11.3.3 TEXT DELETION COMMANDS

nD

Delete n characters.
If n is neg. the n characters preceding pointer are deleted.
If n is omitted, the character immediately after the pointer is deleted.
If n is pos. the n characters after the pointer are deleted.

nK

Kill n lines
If n is omitted, the content of the buffer from the pointer to and including the next <CR> is deleted.
If n is neg. the n lines preceding the pointer are deleted.
If n = 0, the content of the buffer from beginning of line up to the pointer is deleted.
If n is pos. the n lines following the pointer are deleted.

HK

Deletes the entire contents of the buffer. i.e. for small files, deletes everything. Useful to clean out the old file ready to input a new one.

11.3.4 INSERTION COMMAND

Ibaloney$

Inserts the string ' baloney ' in front of the pointer.
The pointer does not move.
If a line is to be inserted, remember the <CR> at the end.

11.3.5 SEARCH (AND CHANGE) COMMANDS

Sblabla$

Searches for the next occurence of the string ' blabla ' in the current text buffer.
The pointer is placed immediately after the string if it is found,
If it is not found, pointer is re-positioned at beginning of text buffer.
Subsequently, the same string can be searched for by just saying S.
If the command is preceded by n, (e.g. nSbean:), it looks for the n'th occurence of the string ' beans '.

Nblabla$

As before, but searches across pages. Don't use this if not necessary as an un-successful search leaves pointer off the end of the file and one has to exit and start over again.

FSblablu

Replaces the next occurence of ' bla ' by ' blu '.
If it can't find ' bla ' it will give an error and pointer is re-positioned at top of the page.
Characters can be deleted from mid line by saying, for example, FSbullshit$$, and ' bullshit ' is removed, or more exactly replaced by nothing.
The pointer is left immediately after the string.
If the command is preceded by n, the n'th occurence is changed.

FNchalk$cheese$

As above but it will search across pages.
If 'chalk' is not found, the pointer is positioned off the end of the file, so you must exit and start again.

11.3.6 EXITING FROM TECO

EX$$

Closes input file, moves current buffer into output file, and exits from TECO, returning to the monitor.

CTRL-C$$
Used to abandon a TECO. It closes input file and kills both the output file and the content of the buffer.

11.3.7 SOME USEFUL Q-REGISTER COMMANDS

^Uqstring$$
This inserts ' string ' into the text storage area of the Q-register 'q'. A Q-register can have any ascii label. The character '^' is the uparrow found above the '6'.

@^Uq/string/$$
This is equivalent to the above command, but 'string' can contain ESCAPE (or ALTMODE) characters.
Useful to store a complex command string which can then be executed simply, and at will. See below.

nXq$$
Moves n lines into the Q-register 'q'.
If n is pos. the .. lines following the pointer are stored.
If n is neg. the n lines before the pointer are stored.

*Q
The previous command string is entered into the Q-register 'q'. This is useful to recover a faulty input string for example. You can then get the string back - see below.
Note though that the stored string will contain two ALTMODES at the end and the faulty commands at the beginning.
Note also that ALTMODES are not needed to terminate this command.

Gq$$
Inserts the content of the Q-register 'q' into the buffer in front of the pointer.

Mq$$
Executes the content of the Q-register 'q'.
Used in connection with the @^Uq/string/ command.

11.3.8 COMMAND LOOPS

n<command string>$$
Executes the command string n times.
If n is not given, it executes the command string indefinately, until it finds an error.

11.4 CREATION OF NEW FILES

Teco is called without giving a filename. When Teco responds with a *, use the command EW followed by the new filename and two ESC'S ($$). This may look like this:

```
>TEC
*EW NAME.EXT$$
```

The new file NAME.EXT is now open and you may continue with normal Teco operations.

11.5 EXAMPLE

We have the file LISHKL.DAT

```
1 1 0
1 3 0
RUBBISH LINE
13 5
-5 7 8
7 -7 7
0 0 0
```

We want

```
1 1 0
1 3 0
1 3 5
-5 -7 8
5 5 6
0 0 16
```

We could do the following (comments in parentheses)

```
>TEC LISHKL.DAT
*HT$$             (type the whole buffer)
1 1 0             (types the file -
1 3 0             don't do this if it's a big file ! )
RUBBISH LINE
13 5
-5 7 8
7 -7 7
0 0 0
*2LT$$            (move 2 lines and type)
RUBBISH LINE      (types new current line)
*K0LT$$           (kill this line and type new current line)
13 5              (new current line)
*FS13$1 3$0LT$$   (change 13 to I 3, type line)
1 3 5             (answer)
*LT$$             (move one line)
-5 7 8            (answer)
*S7$-C$I-$0LT$$   (look for 7, back one character
                          insert '-', re-type the line)
-5 -7 8
*LT$$             (move a line)
7 -7 7            (new current line)
*KT$$             (kill and type the new current line)
0 0 0             (new current line)
*I5 5 o           (insert 5 5 6 <CR>)
$-LT$$            (finish input, type inserted line)
5 5 6
*3FS0$16$0LT$$    (change 3rd 0 to 16, type line)
0 0 16
*J100T$$          (listing of file)
1 1 0
1 3 0
1 3 5
-5 -7 8
5 5 6
0 0 16
*EX$$             (exit)
```

If, on trying to exit from TECO, you get the message ' DEVICE FULL ', you have filled the disk on which you are writing.
DO NOT give any more TECO commands before having made some room on the disk.
Go to the other terminal and purge as many files as possible except, of course, those which you are currently editing. This is important as you may have lost your latest version in the TECO error.
In this case go back and TECO the previous version which will be unchanged.

CHAPTER 12

DATA TRANSFER

Data may be transferred to the Risoe central computer B7800 via a telephone line. The B7800 CANDE system is used to receive data from the PDP-11. Data may also be transmitted from B7800 CANDE to PDP-11.

The PDP-11 program that takes care of the data transfer is called CANDE2.TSK and is started from one of the indirect command files B6700.CMD or B6700A.CMD, depending on whether 1200 baud or 300 baud transmission is wanted.

The connection to B7800 may be set up from any of the PDP-11 terminals. When the line is established the whole range of CANDE commands may be executed from the terminal. A manual with the CANDE language is placed at the spectrometer.

In addition to the CANDE commands five special commands are available.
These are

```
:TTA S FROM <PDP-11 filename>
     -send data from the specified
     PDP-11 file to a CAMAC workfile

:L TO <PDP-11 filename>
     -send data from a CANDE workfile
     to the specified PDP-11 file

:TT
     -specify slow terminal

:LA120
     -specify fast terminal

:HELP
     -list the special commands
```

To use these commands type a ':' on the terminal. The response

on terminal should be

COMMAND :

Thereafter type the rest of the command followed by a carreturn.

A data transfer seqence is shown in the example below.

Example on data transfer to B7800

```
>@B6700                              ;  Call start command file
>;
>;      SETUP FOR CANDE ON TT2:  - 1200 BAUD
>;
>SET /SPEED=TT2:1200:1200
>SET /FDX=TT2:
>SET /RPA=TT2:
>SET /FDX=TT0:
>RUN CANDE2
CONNECT MODEM TO B6700/CANDE
                                     ;  Now dial up CANDE and
                                     ;  establish the line
                                     ;  (use extension no.  4206
                                     ;  for 1200 baud and no.
                                     ;  2121 for 300 baud)


TYPE CR                              ;  When the line is
                                     ;  established
                                     ;  type carreturn on the
                                     ;  terminal
#B7800:128 CANDE 31.280;  YOU ARE TLF/2247(4)
#ENTER USERCODE PLEASE
MHNIELSEN FLINT                      ;  Make yourself known
                                     ;  to CANDE
#ENTER CHARGECODE PLEASE.
150102                               ;  Give the charge code
SESSION 0015 15 15:07:30 05/07/81    ;  CANDE has accepted
                                     ;  your usercode etc.
                                     ;  and is now ready.
MAKE FINN01 DATA                     ;  Specify a CANDE workfile
#WORKFILE FINN01:  DATA              ;  Accepted by CANDE

:                                    ;  Type :
COMMAND :LA120                       ;  Specify fast terminal

#LINE = 120, PAGE = 23, BUF = 0, SCREEN, CONT., "\"
                                     ;  Accepted

:                                    ;  Type :
COMMAND :TTA S FROM DL1:[210,210]FINN01.COL;1
                                     ;  Set up for data transfer
                                     ;  from file FINN01.COL;1
                                     ;  on UIC 210,210 on DL1


#OK

#BEGIN TRANSMIT                      ;  Data transfer starts
#END-OF-DATA                         ;  Data transfer completed
```

```
$                                       ; Wait for that.
                                        ; More files may be send
                                        ; and CANDE commands
                                        ; executed

$LIST 1-300,END                         ; Writes lines 1-300 and
100                                     ; the number of
200                                     ; the last line
300                                     ; of the transmitted data.
41400                                   ; Number of the last line

$LIST 41200-END
41200                                   ; Writes last lines
41300                                   ; of transmitted data
41400
$SAVE                                   ; Save transmitted data
$                                       ; A new CANDE workfile
                                        ; can be specified and
                                        ; data transferred
$WRITE                                  ; Workfile is printed out
                                        ; (to be collected at B7800
                                        ; computercenter under
                                        ; heading FLINT)
BYE                                     ; Exit from CANDE
$END SESSION 0096 ET=3:22.7 PT=0.1 IO=0.1
$USER = MHNIELSEN = 150102 15:37:12 05/07/81
                                        ; CANDE signs off
                                        ; Type CNTR(C) to leave
                                        ; the PDP-11 transmission
                                        ; program
SIGN OFF ???                            ; Remember to sign off
                                        ; CANDE. We have allready
                                        ; done that with BYE
>;      CANDE FINISH                    ; Another CNTR(C) ends
                                        ; the PDP-11 transmission
                                        ; program
>@ <EOF>
>                                       ; NOW BACK IN MCR
```

The transfer rate is 120 characters pr. second, i.e. approx 1 block pr. 5 sec. The number of blocks in the file to be transmitted can be seen, when doing a DIR command.

The data transfer program was written by J.Bundsaard and J.V.Olsen.

CHAPTER 13

RESTART AFTER CRASH

A complete system crash is unusual, and will normally mean a hardware fault. If the system does crash, please inform M.H.NIELSEN, P.LEBECH or F.KREBS LARSEN.

A bootstrap of the system can often be regarded as the ultimate way of getting out of a nasty situation. In this case, firstly, compare your problem with some of the examples given in the following chapter ' TROUBLE-SHOOTING '. If nothing there helps, then bootstrap. Be careful, however, as files can be lost although this would be unusual.
In any case, write down rough details of the problem so that we can, if possible, take steps to avoid it in the future.

Most of the following procedure will probably be unnecessary to clear your particular problem, but until you feel that you know the system well enough to recognise certain symptoms, follow blindly the instructions, and you can't go wrong.

13.1 BEFORE BOOTSTRAPPING THE COMPUTER.

Ensure that power is supplied to the computer and to the camac crate.

Check that disks are running and 'READY'.

Check that terminals are on and 'ON-LINE'

Then bootstrap the system

13.2 BOOTSTRAPPING

Press CNTRL and HLT/SS.

Press CNTRL and BOOT.
(Decwriter gives four numbers)

Type DL <CR> at the Decwriter.

The system will now go through the startup procedure, see below.

13.3 STARTUP PROCEDURE.

You are required to give the time and date in the correct format,
e.g. 15:45 30-SEP-82 . (If this gives a syntax error enter TIM 15:45 30-SEP-82 after the startup.)

Then call LSD.
Restart measurement.
Restart anything else you want running.

****** NOTE ******

The startup procedure sets the dummy disks as follows:

DL1:=DI1:

DL1:=TR1:

DL1:=DR1:

You must re-assign this if necessary, i.e. if you're doing something non-standard like writing data on DL0: or transferring from DL0:

CHAPTER 14

TROUBLE-SHOOTING

This chapter gives a series of examples of common problems and how to get out of them

14.1 NO RESPONSE FROM TERMINALS

The computer is either totally blocked, or stopped.
A stopped computer is recognised by a specific address being shown on the keyboard display rather than a line of zeroes.
Also the ' RUN ' light will go out.
There is not much that you can do in this situation except to bootstrap the system.
Check, obviously, that power is present in all necessary places, that the disks are running and ' READY ',
that terminals are on and do work - try them in local for example - .

When you have the system back, check that there is room on the disks by giving the command
PIP DL0:/FR
to see how many blocks are free on DL0:, for example.
If there is very little room left on DL0: this may be your problem.
Purge f .es as necessary.

14.2 NO RESPONSE FROM TERMINALS EXCEPT MCR>

If a CTRL-C gives a response, but any program called will not work, then the problem is probably one of the following :

DL0: is not ready. Since this is the system disk it can't tell you. It will give a CHECKPOINT WRITE ERROR when it is ready.

You have too many jobs running at once :
Try an ACT /ALL
If it works, abort jobs which are least necessary and the system should start to clear itself.
If ACT /ALL does not work, you will have to remember the state of the system, and abort jobs that you have called.

Your system disk is full :
If it really has zero blocks left you will probably have to bootstrap to be able to purge files, as the system blocks completely.
If it has still one or two blocks left, you can abort jobs to bring the system back and then purge files as necessary.

14.3 A JOB CALLED WILL NOT RUN

If a job is called and it just sits there doing nothing, it is probably too big to get into the core available.
Check that there are no unnecessary jobs running by using the command ACT /ALL

Jobs can be aborted from any terminal.

If you try to run a job which is already running, either the system will give an error saying TASK ACTIVE, or it will give the task name the terminal name as an extension and run it as usual.
For example , a second TECO called, from TT2: (fast modem), will be called TECT2 by the system.

It is not advisable to have more than one LSD running at the same time. In any case NEVER change parameters of a second LSD.

14.4 ERROR IS GIVEN IMMEDIATELY AFTER A JOB IS CALLED

If a job gives a LOAD FAILURE immediately after calling it, the chances are that the disk on which it is kept is either not mounted or not running. Otherwise there is a software or hardware fault.
The command DEV gives the current situation of all devices attatched to the system.
Mount disks with the command
MOU DL1:/OVR
for example. The command
DMO DL1:
similarly dismounts a disk.
A change of disk necessarily means a dismount and mount but running a disk down and up leaves it mounted and still readable by the system, as long as it is ' READY '.

If the message 'TASK NOT IN SYSTEM' is given, it means that the task file is not installed. Search using the command WHE for the TSK file and then install it. An example :
RUN HKLZIG
System responds with ''task not in system.
Do
WHE HKLZIG.TSK
System tells you that this file exists on DL1: in area [200,200].
So now install it as follows
INS DL1:[200,200]HKLZIG
The job should now run.

If, immediately after calling a job, the error message
OPEN FAILURE is given, this means that the file which the program wants to read from is LOCKED. This is caused, normally, by aborting the program while it was reading from this file.
Unlock the file with the command
OPN RAFIN.DAT for example.
The same applies for the error message
'File was not properly closed' in TECO.

14.5 ERRORS SPECIFIC TO LSD

Before taking action, ensure that you do have a genuine error. Remember that LSD is very intolerant to typing mistakes.

Avoid doing ABO LSD if possible - it may cause you to lose files.
Unfortunately sometimes there is no choice as PCP only works at specific times.

14.5.1 FILE RESTAR.LSD NOT FOUND ON DISK

You have done an ABORT of LSD and been unlucky, - the named file is empty or has disappeared from the disk.
Recover a version of the offending file by doing, for example, the following

DEL DL0:RESTAR.LSD
(this may give an error, i.e. the file was really lost and not just emptied), then type

COP DL0:RESTAR.LSD=DI1:RESTAR.BAK
This copies across from DI1: - the data disk - a version of the file from the last time that the disk was initialised.
Check that you have all the files you need by giving the command

DIR DL0:*.LSD;*
This should give THREE files only, ALL should be version No. 1.

PARA.LSD;1
RESTAR.LSD;1
BUTEES.LSD;1

Now call LSD.
Before starting the measurement you must modify various parameters.
For example NUM has the value from the beginning of the disk.
*** This MUST be changed or you will overwrite data. ***
Also the RST command will not continue the measurement from the correct position.
In any case CHECK ALL your parameters and watch the first few reflections after the crash.

14.5.2 LSD IS NOT DOING ANYTHING

This could be one of many things. Try some of these :

Check that the beam is open and make sure the reactor is running.

Ensure that the terminals are not blocked.
Any character(s) typed without a <CR> block(s) the terminal for all output.

Any job which calls the CAMAC, disables the CAMAC for all other jobs. The CAMAC is not re-enabled until that job is terminated. Ensure that you do not have another LSD or CENTRA running out of view.

If you are generating HKL's, remember that limits in H K L which are a long way outside the Theta limits will waste a lot of time in useless generating of unmeasurable reflections. Keep your HKL limits as tight as you dare around the Theta limits.

If you want to read from a list and have not set MIN and PIN properly, see appendix A, LSD may do some strange things including giving a line of garbage and then stop.
In this case you will have to abort and try again.

14.5.3 CAMAC-MOTORS STOPPED MANUALLY

This message is given when the machine runs into a hardware limit. Ensure therefore that your software limits are inside hardware limits.

CHAPTER 15

BUILDING YOUR OWN PROGRAMS

15.1 USING FORTRAN

The computer uses FORTRAN-IV. A manual is available which contains details of the language, commands and functions that are permitted on this system and a list of all possible error messages with their explanations.

Make, using TECO, your program in a file with extension .FTN

Things to note :
The terminal is stream 5 for both input and output.
The commands TYPE and ACCEPT can also be used for interactive communication with the terminal.
Streams 2,3 and 4 are assigned automatically, to files FOR002.DAT, FOR003.DAT, FOR004.DAT respectively for both input and output. It is not advisable to read and write to the same file.
Any other stream must be assigned using the command
CALL ASSIGN (stream no.,'FILENAME.EXT')
Followed by, at the end of the program
CALL CLOSE (stream No.)
The device where these files are to be found is DL0: by default. It can be re-assigned, either in the filename above or in the task builder stage.

When you are happy with the FORTRAN SOURCE file try to compile it.

15.1.1 THE FORTRAN COMPILER

The compiler (FOR>) takes the FORTRAN (FTN) file and produces an OBJECT (OBJ) file and a LIST (LST) file.

Call the compiler with FOR <CR>

```
FOR>PROG,PROG=PROG
```
Will produce from the latest version of PROG.FTN, a file PROG.OBJ for the task builder and a file PROG.LST which contains the program with line numbers, diagnostics and details of all the variables used, functions called etc. etc.

Equally

```
FOR>PROG=PROG
```
Produces just the OBJ file from the FTN file.
This is for confident programmers.

```
FOR>,PROG=PROG
```
Produces just the LST file from the FTN file.

```
FOR>DL1:PROG,DL1:PROG=DL0:PROG
```
Will produce both files on DL1: from the FTN file which is on DL0:.

```
FOR>CTRL-Z
```
Terminates the compiler and brings you back to the monitor.

15.1.2 THE TASK BUILDER

This takes the OBJECT file (OBJ) and builds a core image TASK (TSK) file which is the runable program.

```
TKB <CR>                ( calls task builder )
TKB>PROG=PROG           ( build PROG.TSK from PROG.OBJ )
TKB>/                   ( always )
ENTER OPTIONS :         ( answer )
TKB>LIBR=FORRES:RO      ( always )
TKB>ASG=DL1:2:3         ( optional line, see below )
TKB>//                  ( start task builder execution )
> comes back with ( > ) when finished
```

If this is successful, you will have a file PROG.TSK which is ready to be installed and run.

If you have more than one unit from which to build the task, the second line could have been, for example
TKB>PROG=PROG,SUB1,SUB2,DL1:SUB3

The line
ASG=DL1:2:3
Assigns streams 2 and 3 to DL1:
If you do not give the line, streams 2 and 3 will default to DK0:

15.1.3 INSTALLING THE TASK

Details of the task should be recorded in core for the RUN command.
This is done with the command, for example
INS PROG
or
INS DL1:[200,200]TOTO

The command
RUN TOTO
will then execute the task TOTO.TSK on DL1:[200,200]

If you build a new TOTO.TSK, you must do
REM TOTO and
INS TOTO
which will record details of the new TOTO.TSK -- otherwise the previous TOTO.TSK will be executed.

A bootstrap removes all jobs installed since the previous bootstrap so you will have to re-install them.

15.2 USING BASIC

The BASIC available on the system is a fairly crude version and is not recommended for execution of programs.
It does however, have the advantage that it can be used to do simple, on-line, calculations.

```
RUN BASIC

RSX BASIC                 (answer)

READY                     (answer - ready for command)
RUN DK0:XXX               (executes program XXX.BAS on DK0:)

READY                     (answer - ready for command)
LET A = 550
LET B = 3.8
LET C =  SQR(A/B)
PRINT C
 12.03066                 (gives C)
STOP                      (end program)

READY                     (answer - ready for command)
CTRL-Z                    (stops BASIC and returns to monitor)
```

APPENDIX A

SUB SECTIONS OF LSD

This is a fairly detailed description of the various sections of LSD. A much more detailed manual is available at the machine (in french), but, with luck, you will never need it.

INDEX, see p.A-18

A.1 SECTION CAM

The section CAM is used for all simple CAMAC functions. It is used for such commands as: read encoders, position shafts, do aquisitions, do simple scans.
It is called from the '*' (first level) by typing CAM.
The system will then answer with a 'C'.

A.1.1 READING AND POSITIONING SHAFTS

To read a shaft, e.g.PHI, do PPH <CR>

To position a shaft, e.g. PHI to 90, do PPH:90 <CR>

PTH
Position (of) 2Theta (i.e. detector).

POM
Position (of) Omega.

PCH
Position (of) Chi.

PPH
Position (of) Phi.

PPP
Position (of) all four angles (2Theta, Omega, Chi, Phi)
[eg PPP:50 30 0 -30]

A.1.2 SIMPLE SCANS

Here the format is always of the form

SOM:19 21 .05 30000 0 1 <CR>

i.e. scan from 19 to 21 with step .05, counting 30000 on monitor (= 0 ; or 1 = time , 2 = detector).
The final 1 says write scan details as data on disk. (0=don't)

If you then say SOM again, it will make the same scan again, except that it takes the scan details from SPS in PAR.
This is useful if you want to do the same scan many times.

SOM
Scan omega.

STH
Scan 2Theta, give real 2Theta values.

STT
Scan Omega-2Theta, give Theta values.

SCH
Scan Chi.

SPH
Scan Phi.

A.1.3 MISCELLANEOUS

ACQ
e.g. ACQ:300 1 says count, at this point, for 3 secs .
(0=monitor, 1=time, 2=detector)
NOTE. time is always given in hundredths of a second.

END
This finishes the section CAM and puts you back to first level ('$').

A.2 SECTION PAR

Typing PAR <CR>, when you have the star of the first level, sets you into the section PAR. The terminal will respond after each command with a P.
This section contains all the parameters for the running of LSD.

To read a group of parameters, e.g. MRE - Your resolution curve, Type MRE <CR> and the set of parameters will be outputted.

To change, for example the fifth parameter to 25, type
MRE:= = = = 25 <CR>

To get back to the first level of LSD, Type END<CR> and the terminal will respond with a *.

NOTE

The section PAR uses and updates the files PARA.LSD, BUTEES.LSD, RESTAR.LSD. In the case of serious computer crashes, these files can be lost. For this eventuality, the files are written, as a backup, on the data disc at the time of initialisation. If you find yourself without any of the files - LSD will make it quite plain what is wrong - you must do, for example -

```
DEL SY:RESTAR.LSD
COP SY:RESTAR.LSD=DI1:RESTAR.BAK
```

This then copies back the file to where LSD can use it. Remember that the file will need modification as it is from the time of the previous disk change. If you are lucky, it will only be NUM that needs changing.

A.2.1 PARAMETER LIST

MCO

Comments

To change type MCO:<CR>, then enter new text.

MUT

Name of experimentalist and local contact.

Two times six characters for these two names. e.g. ALIBONLEHMAN

To change type MUT:<CR>, then enter new text.

WAV

Wavelength

LAT

Lattice constants. (Real cell).

Needed for IND command for example.

MUB

Orientation matrix, Busing and Levy convention.

When changing, always ensure that it has been entered correctly.

MRE

Resolution curve

Used when IDELTA in SPS is set to 1.

Number of points - MAX = 5
Followed by the angles - THETA.
Followed by the respective values of
total scan width in Theta.

ZER

Machine zero offsets

Do not change without consulting M.H.Nielsen, B.Lebech or F.Krebs Larsen.

If ZER is changed then the display shows the wrong angles until the program DISPLA has been stopped (use ABO DISPLA) and restarted (use RUN DISPLA).

In the order : 2THETA, OMEGA, CHI, PHI.
Note- Crystal angle = Encoder angle - 180.00 + Offset

SUB SECTIONS OF LSD PAGE A-6
PAR

BUT
Machine software limits

Do not change without consulting M.H.Nielsen, B.Lebech or F.Krebs Larsen.

The software limits must always be inside the hardware limits.

Values given are observed values minus values found in ZER.

MIN
H K L Generation rules

IZIGZA : 0 = H K L Given in file (see also PIN).
1 = H K L Generated internally by LSD.
2 = H K L 2THETA OMEGA CHI PHI Given in file.
NOTE: For IZIGZA = 1 set IE = 5 in PIN.
IHGEN)
IKGEN) : Relative speeds of variation. 1=Fastest, 3=Slowest
ILGEN) IHGEN + IKGEN + ILGEN = 6 ==>ILGEN NOT GIVEN
(SUPERFLUOUS).

Then : Neg H Limit, Pos H Limit, Neg K Limit,
Pos K Limit, Neg L Limit, Pos L Limit.
Don't give wider limits than necessary.
Generation can waste a lot of time.

Then : Three Nos. for elimination of H<K<L for example.
Normally 0 0 0. Rarely used.

NUM
Number of previous reflection measured by LSD.

Do not change unless absolutely necessary !! (but must be changed if RESTAR.LSD is lost and an old version is recovered.)

MEX
Extinction rules

Nine numbers, see appendix B.

SPS
Scan details (see also SCA p.A-7)

CONS : Preselected No. of counts for each point on :--
JCODE : 0=Monitor, 1=Time, 2=Detector.
IDELTA : Fixed scan width ? 0=Yes, given below.
1=No, Calculated in MRE.
DELTAV : Scan width if IDELTA = 0.
(If IDELTA=1, This is the scan width calculated by MRE)

SLI

Limits on theta (not 2-theta) for scans.

Used if you want to collect data in shells. Must be inside the hardware limits, including scan width.

SCA

Scan details (see also SPS p.A-7)

LCODE	: Usually either 2 = Omega-scan or 5 = Omega-X Theta scan.
NK	: Number of points fitted in scan width.
IX	: For coupling factor,- 0=Given, 1=Calculated.
X	: If IX=0, this is fixed coupling value. (If IX=1 a coupling curve is given (5 Points))
THETAX	: The five angles.
XX	: And the five corresponding values.

MBG

Separate steps in background.
Use only when you want e.g. long steps outside peak to ensure setting down to background, or short steps to avoid running into next peak.

No. of points on each side.
Step length between points.
Monitor count (Usually same as in SPS).

MTE

Test reflections.

Number of test reflections.
Frequency of test reflecctions.
Start with test refls after MES ? 1=Yes, 0=No.
Followed by H K Ls for up to four test reflections.

WARNING: The test reflection is measured for preset Monitor.
Preset Monitor is given in CHA 335.
To change Monitor e.g. to 10000 type
CHA: 335 1 10000 <CR>

CRY

Parameters for when you're using a cryostat.

ICRYO : 0/1 For without/with cryostat.
IANAL : 0/1 For without/with analyser. (usually 0).
ISN : To position PHI.
= -1 Always arrives turning negative.
= 0 Takes shortest route between points.
= 1 Always arrives turning positive.
= 2 Does not go through 180 Deg. (Obligatory if you're using a cryostat).
IVOL : 0/1 For don't read/read thermocouple voltage at end of each scan. For this to work, the switches in the section MESURES should read A1, B1, C1.
CIMINI : Min CHI value. Separate test to that in BUT. Normally set to more neg. value than that of BUT.

PIN

Input of H K Ls for MES. (see also MIN)

IE : Mode of entry ,
= 1 For when you are generating HKL's.
= 2 Via a file called FOR002.DAT.
= 3 Via a file called FOR003.DAT.
= 4 Via a file called FOR004.DAT.
= 5 Direct by keyboard, finish with CTRL-Z.
IRESTA : 0/1 For non-authorisation/authorisation of restart.

CEU

Constants for centering routines.

ITYPE : 0/1 For centering by Peaks/Half-shutters.
ICYCE : Number of cycles. (Unlimited but usually 2 or 3).
ISCCE : 0/1 For No/Yes to Omega scan at end of centering.
NPTS : Number of points for centering scans.

CED

Parameters for centering routines.

GRANDD : Distance between sample and detector.
PETITD : Detector aperture in cm. (Make it 50% larger)
COURB : Three angles in Theta,
Followed by their respective scan widths.
(This is for centering of Omega only - not for MES.)

CET

Parameters for centering routines.

RATIO : eg 0.5 ->Max is in center of half heights.
CPERL : eg 0.1 for 10% difference in half-shutter readings.
AMEIL : Scan width AMEIL * COURB at second scan attempt.
COUMIN : Minimum counts for a peak.
TENMAX : Maximum counting time for a point.
COUNOR : Normal counting time for a point.
PARTTH : Max error between cycles on 2Theta.)
PARTO : Max error between cycles on Omega.) Conversence
PARTC : Max error between cycles on Chi .) Criteria.

MOU

Scan output details.

ISOR : Output format for counts.
= 0 No output of counts at teletype.
= 1 Outputs 10 counts per line.
= 2 Outputs angle and count line by line.
= 3 As 2, plus a star to represent profile.
Logaritmic scale
NOTE: MOU = 3 can not be used for
simple scans (see A.1.2, P.A-2)
IVISU : 0/1 For don't display/display scan on screen.
(MAKE SURE SCREEN IS CLEAR OR ELSE THE SYSTEM BLOCKS)
IF2 : Rarely used.

MIL

Current H K L being measured

Rarely used, HKLs for restart are stored elsewhere.

MPA

Miscellaneous parameters for scans.

IPARA : 0/1 For bisecting/parallel geometry.
IBLIND : In cases where a reflection is outside limits,-
= 1 Go to next one.
= 4 Does -H -K -L.
= 6 Goes to 180 - Chi , 180 + Phi.
Do not use IBLIND = 6 Without consulting one of us experts.
NORM : Monitor count to which FSQ. is scaled.

CONSM : (7th parameter) Monitor count for a fast scan
to check whether a reflection is observed in
OPT2 and OPT3, (see P.A-15)

SKNM1 : (8th parameter) Limit for observed reflections
in OPT2. We measure four points in the peak.
The first point (C1) at the beginning of the scan,
the second (C2) at 3/8 of the scan, (C3) at 5/8
of the scan and (C4) at the end of the scan.
If sigma (I)/I = SQRT(C1+C2+C3+C4)/(C2+C3-C1-C4)
is greater than SKNM1, then the reflection is
skipped.
Typical SKNM1 is around 0.75.

A.2.2 THE COMMAND CHA IN PAR.

Some parameters which are changed rarely are only accessible using the command CHA. There follows a short list of the ones that you are most likely to come up against.

To read a value, eg Parameter No. 362, Do-

CHA 362 1

To change parameter 362 to 5000, do-

CHA:362 1 5000

The terminal will confirm by tellin you which parameter you have changed and to what.

Parameter 53
Character used to display points on screen.

= 0 Character is '.'.
= 1 Character is '+'.
= 2 Character is '*'.

Parameter 79
Language of errors, 1 = french
2 = english
3 = german.

Parameter 85
Steps rounded off to nearest hundreth in scans.
0 = Yes (eg steps of .020)
1 = No (eg steps of .0233)
Normally we use 1, it saves time.

Parameter 362
Scale of display on screen. Y direction

Parameter 335
Preset value for Monitor for test reflections
To change Monitor ex. to 10000, type
CHA: 335 1 10000 <CR>

A.3 SECTION CAL

CAL is for calculating angles.

CAL0 H K L

Calculates from MUB and WAV the angles for H K L.

CAL0 H K L PSI
As above, but with a the PSI angle specified. Will no longer be bisecting.

CAL4
Calculates H K Ls as in MES. Will zigzag according to MIN and SLI. Stop it with PCP = 3 when you have seen enough.

CAL5
Calculates H K Ls from a file FOR00X.DAT Where X is given in PIN.

CAL6 H K L PSIMIN PSIMAX PSIPAS
Calculates angles for HKL with PSI varying from PSIMIN to PSIMAX in steps of PSIPAS.
Useful to do before a Renninger scan, see REN0, to see what is allowed by machine limits.

A.4 SECTION HKL0

HKL0 H K L (PSI)
Calculates angles and puts machine there, if it can.

A.5 CENTERING ROUTINES.

CEN0 H K L
Calculates angles from MUB and centers from that point.

CEN2 H K L 2Theta Omega Chi Phi
Centers from these angles. H K L has no importance here but should still be given.

CEN4 H K L
Reads current machine angles and centers from this point. H K L has no importance here, but should still be given.

CEN5
Centers from H K Ls given in FOR00X.DAT where X is given in PIN.

A.6 COMMANDS INV AND IND1

IND1 H K L
Calculates D-spacing and Theta angle for H K L from WAV and LAT only. Useful when searching for reflections.

INV 2Theta Omega Chi Phi
Inverse of CAL0, using MUB and WAV, it gives you the H K L corresponding to these angles. (assumes bisecting)
Useful when trying to index crystal faces for example.

A.7 COMMAND XBU

The command XBU causes LSD to accept its commands via a buffer. The file used as the buffer is called COMAND.LIS on DL0:.
Therefore if, using TECO, you put in the file the series of command that you want executed and then type XBU in LSD, these commands will be executed as if you were entering them on the terminal. At the end of the file a message is given, and the input stream is given back to the teletype.

Example

The file COMAND.LIS might contain

```
PAR
PIN:2
MCO:
FIRST DATA SET - STRONG REFLNS
SPS:20000
MBG:= = 20000
END
MES
PAR
PIN:3
MCO:
SECOND DATA SET - WEAK REFLNS
SPS:100000
MBG:= = 100000
END
MES
```

A.8 MEASUREMENT ROUTINES

MES

When you are happy with your orientation matrix, and have set such things as scan widths, detector apertures, etc. etc. you will normally start the measurement using the command MES.

Before starting, LSD checks whether the beam is open or not. If not, it gives you a message. The check can be overwritten (if the reactor is down) with a 0.

The first thing LSD will do is to see if it should start with the test reflections - thereby initialising their values for subsequent checking. Once it has finished this, it will look at IZI6ZA in MIN to see whether it should generate refns or read from a list.
It will then measure HKLs assuming they are inside software limits and also specified Theta limits. When it has finished the series, it will come back with the '*' of the first level.

If you want to interrupt it, you should use PCP -see next section- and then restart - see RST.

RST

This command automatically causes measurement of the last HKL started, then the measurement carries on from that point.

If you are measuring from a list, you must exit from LSD and then re-enter before doing RST.
The HKL being measured is remembered by recording the input file pointer position. This is written in the file RESTAR.LSD so if you touch your input file, e.g. copy it, TECO it etc. RST will no longer work. Equally, if you have lost your file RESTAR.LSD and bring back an old version, RST will not work.
In any case, you should watch the first reflections after a RST to check that all is well.
If RST will not work, you must stop LSD, TECO out the HKL's that you have already measured and do a MES to start at the top of the new list.

If you are generating, your current position is remembered by recording, in the file RESTAR.LSD, the current HKL and the directions in which H, K, L are ziszasing. These Nos. can get scrambled, although this is rare, watch the first reflections after a RST to check that all is well.

You can restart from a specific HKL (if you are generating) by typing : RST H K L and then the directions in which HKL

are zigzagging. e.g. --

RST 10 -4 12 -1 1 1
(H was going negative, K and L pos.)

OPT2 or OPT3
There is a facility where you can quickly scan a peak using (OPT3) or (OPT2) points to see if it is worth measuring. Obviously this will save enormous amounts of time if most of your reflections are weak.
See MPA, p.A-9 for details.

REN0 H K L PSIMIN PSIMAX PSIPAS
This enables you to do Renninger scans, i.e. turning around the scattering vector. You specify from PSIMIN to PSIMAX with the desired step in PSI.
The machine then measures if it can. Remember the machine goes out of bisecting unless CHI is exactly 90 deg.
REN does a complete step-scan for each specified value of PSI, so the same parameters for MES are needed for REN.

PSI
Does genuine Renninger scans, i.e. measures only one point at each value of PSI. Useful to check multiple scatterings.

QSC
This command enables you to do Q-scans, any scan along a specified straight line in reciprocal space.
You give min and max HKL values and the No. of points.
The machine is positioned at the minimum HKL and a count is made. HKL is then incremented according to the number of steps asked for and the min and max values.
It does No. of points.

Called with : QSC POINTS HMIN HMAX KMIN KMAX LMIN LMAX

Example : QSC 10 .8 1.3 .8 1.3 0 0

Measures the following sequence, one point at each HKL :

0.8	0.8	0
0.85	0.85	0
0.9	0.9	0
0.95	0.95	0
1.0	1.0	0
1.05	1.05	0
1.1	1.1	0
1.15	1.15	0
1.2	1.2	0
1.25	1.25	0
1.3	1.3	0

Beware, since you are doing only one point at each HKL, your UB matrix must be good enough to ensure accurate

defining of your position in reciprocal space.
Your sample and slits should both be small.
The monitor count is taken from SPS in PAR.

A.9 PCP (INTERRUPTION OF LSD)

PCP

This is an external program which permits you to interupt a measurement.
IT MUST BE USED ONLY WHEN THE MACHINE IS COUNTING as it acts at the end of an aquisition.
(It also works during a series of angle calculations (cal) so is used to stop CAL4 for example). Typing PCP runs a program that demands a number from 0 thro' 4.

0 = PCP does nothing, i.e. PCP is not executed.

1 = LSD goes into a pause. It can then be restarted from that point by typing RES LSD. LSD can not be accessed in the meantime.

2 = The HKL being measured is immediately stopped, and the next is started.

3 = LSD returns to the '*'

4 = LSD returns to the '*', and terminates XBU command properly.

As already stated PCP acts at the end of a counting point. Therefore if LSD continues counting, try again.

A.10 INDEX

SUB SECTIONS OF LSD PAGE A-19
INDEX

APPENDIX B

RULES LIMITING POSSIBLE REFLECTIONS

B.1 H K L

0 : No conditions
1 : H + K + L = 2n
2 : H, K, L all even or all odd
3 : -H + K + L = 3n
4 : H = K + L = 3n
5 : H + K = 2n
6 : K + L = 2n
7 : H + L = 2n
8 : H + K + L = 6n
9 : H, K, L all even
10 : H, K, L all odd
11 : If H - K = 3n, then L = 6n

B.2 H K 0

0 : No conditions
1 : H = 2n
2 : K = 2n
3 : H + K = 2n
4 : H + K = 4n

B.3 0 K L

0 : No conditions
1 : K = 2n
2 : K + L = 2n
3 : K + L = 3n
4 : K + L = 4n
5 : L = 2n

B.4 H 0 L

0 : No conditions
1 : L = 2n
2 : H = 2n
3 : L + H = 2n
4 : L + H = 4n

B.5 H H L

0 : No conditions
1 : L = 2n
2 : H = 2n
3 : 2H + L = 4n

B.6 H-H L

0 : No conditions
1 : L = 2n

B.7 0 K 0

0 : No conditions
1 : K = 2n
2 : K = 4n

B.8 H 0 0

0 : No conditions
1 : H = 2n
2 : H = 4n

B.9 0 0 L

0 : No conditions
1 : L = 2n
2 : L = 3n
3 : L = 4n
4 : L = 6n

APPENDIX C

NEUTRON SCATTERING AMPLITUDES

Coherent neutron scattering amplitudes in units of 10 E-12 cm.
Complex amplitudes correspond to Lambda = 1A

G.E.Bacon

Element	Isotope	b
H	1 H	-0.374
	2 H	0.667
	3 H	0.47
	4 H	0.30
He	4 He	0.30
Li		-0.214
	6 Li	0.18+0.025i
	7 Li	-0.233
Be		0.774
B		0.54+0.021i
	10 B	0.14+0.11i
	11 B	0.60
C	12 C	0.665
	13 C	0.60
N	14 N	0.94
	15 N	0.65
O	16 O	0.580
	17 O	0.578
	18 O	0.600
F	19 F	0.56
Ne		0.46
Na	23 Na	0.36
Mg		0.52
	24 Mg	0.55
	25 Mg	0.36
	26 Mg	0.49
Al	27 Al	0.35
Si		0.42
P		0.51
S		0.28

Element	Isotope	b
Cl		0.96
	35 Cl	1.18
	37 Cl	0.26
Ar		0.20
	36 Ar	2.43
K		0.37
	39 K	0.37
Ca		0.47
	40 Ca	0.49
	44 Ca	0.18
Sc	45 Sc	1.18
Ti		-0.34
	46 Ti	0.48
	47 Ti	0.33
	48 Ti	-0.58
	49 Ti	-0.08
	50 Ti	0.55
V	51 V	-0.038
Cr		0.352
	52 Cr	0.490
Mn	53 Mn	-0.37
Fe		0.95
	54 Fe	0.42
	56 Fe	1.01
	57 Fe	0.23
Co	59 Co	0.28
Ni		1.03
	50 Ni	1.44
	60 Ni	0.28
	61 Ni	0.76
	62 Ni	-0.87
	64 Ni	-0.04

NEUTRON SCATTERING AMPLITUDES

Element	Isotope	b
Cu		0.76
	63 Cu	0.67
	65 Cu	1.11
Zn		0.57
	64 Zn	0.55
	66 Zn	0.63
	68 Zn	0.67
Ga		0.72
Ge		0.82
As		0.64
Se		0.80
Br		0.68
Kr		0.74
Rb		0.71
	55 Rb	0.83
Sr		0.69
Y	59 Y	0.76
Zr		0.71
Nb		0.71
Mo		0.69
Tc		0.68
Ru		0.73
Rh		0.58
Pd		0.60
Ag		0.60
	107 Ag	0.83
	109 Ag	0.43
Cd		0.37+0.16i
	113 Cd	-1.5+1.2i
In		0.39
Sn		0.62
	116 Sn	0.58
	117 Sn	0.64
	118 Sn	0.58
	119 Sn	0.60
	120 Sn	0.64
	122 Sn	0.55
	124 Sn	0.59
Sb		0.56
Te		0.58
	120 Te	0.52
	123 Te	0.57
	124 Te	0.55
	125 Te	0.56
I	127 I	0.53
Xe		0.48
Cs		0.55
Ba		0.52
La		0.83

Element	Isotope	b
Ce		0.48
	140 Ce	0.47
	142 Ce	0.45
Pr	141 Pr	0.44
Nd		0.77
	142 Nd	0.77
	144 Nd	0.28
	146 Nd	0.87
Pm		
Sm		
	149 Sm	-1.9+4.5i
	152 Sm	-0.5
	154 Sm	0.96
Eu		0.68
Gd		1.5
	157 Gd	4.3+4i
	160 Gd	0.91
Tb	159 Tb	0.76
Dy		1.69
	160 Dy	0.67
	161 Dy	1.03
	162 Dy	-0.14
	163 Dy	0.50
	164 Dy	4.94
Ho	165 Ho	0.85
Er		0.79
Tm		0.72
Yb		1.26
Lu		0.73
Hf		0.78
Ta		0.70
W		0.48
	182 W	0.83
	183 W	0.43
	184 W	0.76
	186 W	-0.12
Re		0.92
Os		1.07
	188 Os	0.78
	189 Os	1.10
	190 Os	1.14
	192 Os	1.19
Ir		1.06
Pt		0.95
Au		0.76
Hg		1.27
Tl		0.89
Pb		0.94
Bi		0.86

Element	Isotope	b	Element	Isotope	b
Po			U		0.85
At				235 U	0.98
Rn				238 U	0.85
Fr			Np		1.05
Ra			Pu		0.75
Ac				240 Pu	0.35
Th	232 Th	1.03		242 Pu	0.81
Pa		1.30	Am	243 Am	0.76
			Cm	244 Cm	0.7

The values of b given above have been derived by a variety of methods and in some cases their precise accuracy is uncertain. It is pointed out that for nuclei giving resonance scattering, and not merely for acknowledged complex scatterers like Cd, b is not wholly independent of wavelength. The values of b should, therefore, be used with discretion when lambda is not close to 1A.

Published by the Neutron Diffraction Newsletter,
W.B.Yelon, Editor.

APPENDIX D

SYSTEM COMMANDS AND PROGRAMS

This appendix is aimed at giving you a reminder of all the system commands and programs that you may need during your experiment. It also tells you where to find the more detailed description of the command in the manual.

D.1 COMMANDS FOR THE SYSTEM.

RUN XXX
Executes the program XXX.
It looks for a file XXX.TSK which is the task image file of the program. It must be installed but this has normally been done for you.

INS XXX
Installs the program XXX. Essentially, this command records details of the task, it's location etc, in core, so that the RUN command can be executed.
Most tasks will already be installed but if you build your own program, you must do this before executing it.
See chapter 17 for more details.

REM XXX
Not a command that you will normally use.
This removes the task XXX from the task tables.
See chapter 17 for more details.

ABO XXX
This aborts the task (program) XXX
It will work from any terminal.
Avoid doing ABO LSD and ABO ADVENT, you will destroy files.

ACT /ALL
Shows all programs running at the moment. Example
. LDR.
...MCR
...SYS
F11ACP
...LSD
T11V10
...TEC
Showing LSD, T11V10 and TECO are running. The first four tasks are system programs.

ASN DL1:=TR1:/GBL
Assigns the dummy device TR1 to the disk DL1:
See section 2.3.1

ASN /GBL
Gives the current status of dummy disk assignments.
See section 2.3.1

MOU DL1:/OVR
Mounts DL1: ready to be read by the system.
This does nothing to the disk.

DMO DL1:
Dismounts the disk DL1:

DEV
Tells you the status of all devices attatched to the system.
i.e. whether they are loaded, mounted etc.

TIM
Gives computers time and date.
To change give the following command and follow strictly the format.
TIM 10:56 19-OCT-79

HEL
Command to log onto the system.
See section 2.4.

BYE
The command to log-off the system.
See section 2.4

D.2 PROGRAMS ON THE SYSTEM.

LSD
The main program for driving the diffractometer, doing the measurement etc, etc.

It uses the files PARA.LSD, RESTAR.LSD and BUTEES.LSD.
It will read HKL's from FOR002.DAT, FOR003.DAT, FOR004.DAT.
Other files it may use are COMAND.LIS, FOR012.DAT.
See chapter 4 and appendix A.

RAFIN

The program to refine the orientation matrix, cell constants, wavelength and/or machine zeroes from specific reflection positions.
See chapter 5.

INDEX

This is a program to index reflections according to their position in reciprocal space.
It reads from INDEX.DAT and will create RAFIN.DAT for RAFIN.
See chapter 5.

HKLGEN

This is the program to generate a list of HKL's from orientation matrix.
See chapter 6

COLL5N

Data reduction program.
See chapter 8.

LISCAR

Gives the content of the data disk DL1:
See chapter 7.

DATFIR

Gives detailed information about what is stored on disk for a specific reflection number.
See chapter 7.

@B6700

Program to transfer data to B7800, see chapter 12.

CENAUT

This is a command string connecting a series of programs to help you to center and refine the orientation of your crystal. The programs are :
DEFHKL, which writes to DEFHKL.DAT,
CREHKL, which writes to CREHKL.DAT,
CENTRA, which reads from LISHKL.DAT,
and writes to CENTRA.DAT and RAFCEN.DAT,
RAFOTO, which reads from all the files created above.
See chapter 10.

PARAME

Program to give you details of your parameters in PAR.
Gives you exact details of all your parameters.
See chapter 4.6

TECO

The system routine used to edit and correct any file.
See chapter 11.

EW

To create a new file type
>TEC
*EW NAME.EXT$$
*
see chapter 11.4, p.11-6

PCP

The program used to interrupt LSD.
Should always be used rather than ABO LSD as this causes lost files.
See section A.9.

KCO

The program used to stop COLLSN when it is running in automatic so that one doesn't lose data files.
See chapter 8.

NOTE

The programs: LSD, TEC, PCP and KCO
are activated by name only.

The programs: RAFIN, INDEX, HKLGEN, COLLSN, LISCAR, DATFIR, @B6700, CENAUT and PARAME
are activated by RUN followed by the name.

Risø National Laboratory **Risø - M - 2344**

Risø - M - 2344

Title and author(s)

Write-up for the diffractometer D1 at Risø

Edited by J. Bundgaard, F. Krebs Larsen, B. Lebech, M.H. Nielsen and P. Skaarup

Date May 1982

Department or group

Group's own registration number(s)

120 pages + 0 tables + 5 illustrations

Abstract

Manual for the crystallographic program system used to control the 4-circle neutron diffractometer D1/TASII at DR3, Risø. The mechanical part of the diffractometer consists of a monochromator part which allows an easy change of incident neutron wavelength and a four-circle HUBER goniostate consisting of an Euler cradle (HUBER 512) and two horizontal goniometers (HUBER 440 and HUBER 430). The goniostate is computer controlled by a PDP-11/34 interfaced via CAMAC modules. The PDP-11/34 computer has a 128 k byte memory, two hard magnetic disc stations, a fast DEC-writer terminal and a screen terminal. The diffractometer can be operated remotely via modem and telephone line connections from remote stations such as the University of Århus and ILL, Grenoble. Minor parts of the software used to control the diffractometer were developed at Risø while the major parts of the control and data reduction software were a generous gift to Risø from College 5, the diffraction group, at the Institute Laue-Langevin, Grenoble, France.

Copies to

Library (2)

Available on request from Risø Library, Risø National Laboratory (Risø Bibliotek), Forsøgsanlæg Risø), DK-4000 Roskilde, Denmark
Telephone: (03) 37 12 12, ext. 2262. Telex: 43116